

PMRE - Project Management & Requirements Engineering

Zusammenfassung

Maurin D. Thalmann

30. Juni 2019

Inhaltsverzeichnis

1 Project Management - MTW in der Projektabwicklung	3
1.1 Projektarbeit & -Abwicklung	3
1.1.1 Aufgaben des Projektmanagement	3
1.1.2 Aufgaben der Systemgestaltung	3
1.2 Projektmanagement	4
1.2.1 Projektmethoden	4
1.2.2 Grundlegende Denkansätze bei Projektmethoden	5
1.3 Systemgestaltung	6
1.3.1 Systemdenken	6
1.3.2 Systemmerkmale und -Struktur	6
1.3.3 Darstellungsarten	7
1.3.4 Denkansätze und Betrachtungsweisen	7
1.3.5 Systemtypen	7
1.3.6 Anwendung des Systemdenkens im Projektmanagement	7
1.3.7 Dynamik von Systemen	8
1.3.8 SE-Vorgehensmodell	9
1.3.9 Problemlösungszyklus - PLZ	10
1.4 Erfolgsmuster in der Projektarbeit	12
2 Project Management - Projekt in Gang setzen	13
2.1 Projekt präparieren	13
2.2 Zielbegriff	13
2.3 Anforderungsbegriff	13
2.4 Ziel-Mittel-Denken	14
2.5 Zielrelationen (-Matrix)	15
2.6 Zielgewichtung	15
2.6.1 Paarvergleich	15
2.6.2 Präferenzmatrix	15
2.6.3 Analytical Hierarchy Process (AHP)	16
2.6.4 Lieferobjekte aus „Projekt präparieren“	16
2.7 Projekt planen (Planungsprozess)	16
2.7.1 Meilenstein- / Phasenplanung	16
2.7.2 Projektstrukturplanung (PSP)	17
2.7.3 Ablaufplanung	18
2.7.4 Ressourcenplanung	22
2.7.5 Organisationsplanung / Aufbauorganisation	24
2.7.6 Projektkostenplanung	25
2.7.7 Terminplanung	27
2.7.8 Projektbudgetplanung	27
2.7.9 Informationsplanung	28
3 Project Management - Projekt in Gang halten	29
3.1 Kontrollbereiche	30
3.2 Terminkontrolle	31
3.2.1 Termintreue (%)	31
3.2.2 Zeitlicher Fortschrittsgrad (%)	31
3.2.3 Termin-Trenddiagramm	31
3.3 Kostenkontrolle (Aufwandkontrolle)	32
3.3.1 Ist- und Plankosten	32
3.3.2 SOLL-Kosten (Earned Value)	32
3.4 Leistungskontrolle / Sachfortschrittskontrolle	33
3.4.1 Methoden zur Bestimmung des Fortschrittsgrads	33
3.5 Integrierte Kontrollsicht - Leistung-/Kosten-Kontrolle	34
3.5.1 Earned Value (EV)	34
3.5.2 Earned Value Methode (EVM)	35

3.5.3	Beispiel einer Anwendung der EVM	36
3.5.4	Erfahrungssicherung	36
3.5.5	Reporting / Berichtswesen	36
3.6	Projekt steuern	37
3.6.1	„Adding People to a late project makes it even later“	37
3.6.2	Dynamik der Projektarbeit	37
4	Requirement Engineering - Einführung / Setting	39
4.1	Ziel	39
4.2	Stakeholder	39
4.3	RE - Komponenten	40
5	Requirement Engineering - Anforderungen ermitteln	40
5.1	Klassifizieren von Stakeholder-Inputs	40
5.2	Ermitteln	41
5.3	7 Product Dimensions	41
5.4	W-Fragen für bspw. Interviews	41
6	Requirement Engineering - Anforderungen dokumentieren	42
6.1	Drei komplementäre Perspektiven	42
6.2	Sichten der Anforderungsformulierung	42
6.3	Diagramme im RE	42
6.3.1	Beispiele von Diagrammen	43
6.3.2	Modellelemente verschiedener Diagramme	44
6.4	Anforderungen in Textform	45
7	Requirement Engineering - Agiles RE	46
7.1	Dokumentation von Anforderungen in agilen Projekten	47
7.1.1	Blacklog Items in agilem Projekt	47
7.1.2	User Story Mapping	47
7.1.3	Scrum User Story	48
7.2	Scrum Backlog Items	49
7.2.1	Tasks	49
7.2.2	Epics	49
7.2.3	Quality Attribute Scenario	49
7.2.4	Spikes / UX-Spikes / Other Work Items	50
7.3	User Story Splitting	50
8	Requirement Engineering - Qualitative Anforderungen	51
8.1	Anforderungsarten	51
8.2	Vorgehen beim Klassifizieren von Anforderungen	51
8.3	Anforderungstypen ISO/IEC 25030	51
8.4	Quality Model	51
8.4.1	Quality In Use Attributes	52
8.4.2	Product in Use Attributes	52
8.5	Abnahmekriterien	52
8.5.1	Beispiele von Metriken	53
8.6	Quality Grid	53
9	Requirement Engineering - Anforderungen prüfen	53
9.1	Überprüfen von Anforderungen	53
9.2	Reviews - Begriff, Objekte, Typen & Arten	55
9.2.1	Reviews / Audits / User Acceptance Tests	55
9.2.2	Reviews nach Objekten	55
9.2.3	Reviews im Detail	55
9.3	Anforderungsüberprüfung in hybriden Projekten (SoDa)	56

1 Project Management - MTW in der Projektabwicklung

1.1 Projektarbeit & -Abwicklung

Systemgestaltung Ein System um- oder neugestalten, um ein erkanntes Problem (unbefriedigenden Zustand) zu lösen

Problemlösungsprozess (makro) Problemlösungsvorgehen anwenden, welches durch Projektorganisation und prozessuelles Vorgehen (Phasenplan) strukturiert wird

Projektmanagement Projektleitung dirigiert Problemlösungsprozess nach vorgegebener Art und Weise, um Erfolg des Vorhabens sicherzustellen

Problemlösungszyklus (mikro) Projektteam wendet im Laufe der Problemlösung Methoden, Techniken, Werkzeuge an, um ein optimales Zielsystem zu gestalten

Projekt Ein Projekt beinhaltet alle Tätigkeiten, um einen IST-Zustand (Problem) in einen Soll-Zustand (Lösung) zu überführen

Projektabwicklung Projektabwicklung legt den Fokus auf die Art und Weise der Abwicklung eines Projekts, um ein vereinbartes Projektziel zu erreichen. Meist wird eine vorgegebene Art und Weise des Vorgehens für die konkrete Abwicklung genutzt (Projektmethode a.k.a. Vorgehensmodell oder Vorgehensmethode).

Unterscheidung Projektmanagement vs. Systemgestaltung

Projektmanagement Fokus auf das Management des Problemlösungsprozesses, die Planung und Disposition der Ressourcen, die Organisation der Informationsflüsse, der Meinungsbildungs- und Entscheidungsprozesse etc.

Systemgestaltung auch System Engineering; Fokus auf die Problemlösung im eigentlichen Sinne, die effektive Um- oder Neugestaltung eines Systems

1.1.1 Aufgaben des Projektmanagement

Projektmanagement als Sammelbegriff für alle planenden, überwachenden, koordinierenden und steuernden Tätigkeiten, welche zur erfolgreichen Durchführung der Systemgestaltung notwendig sind. Das Vorgehen zur Erreichung der Lösung steht im Vordergrund (nicht das System an sich). Projektmanagement lässt sich in folgende Dimensionen gliedern:

Funktionale Dimension Alle Aufgaben der Ingangsetzung, Inganghaltung und dem Abschluss eines Projekts

Institutionale Dimension dalle Aufgaben, um formale Organisations- und Entscheidungseinheiten zu schaffen, welche im Projekt etabliert werden (Projektteams, Gremien und Rollen)

Instrumentale Dimension Alle Aufgaben der Definition, Einführung und Unterhalt von Methoden und Werkzeugen zur zeitlichen, kapazitiven, qualitativen und kostenbezogenen Planung, Überwachung und Steuerung eines Projekts

Personelle Dimension Alle personenbezogenen Komponenten in einem Projekt, Wahl von Mitarbeitenden, Zusammenarbeit, Kommunikation, Konfliktbewältigung etc.

Psychologische Dimension Alle personenbezogenen Aspekte eines Projekt, bezogen auf Akzeptanz der Ziele, Vorgehensweisen, Verhaltensmuster, Sinnhaftigkeit des Vorhabens durch Mitarbeiter.

1.1.2 Aufgaben der Systemgestaltung

Bei Systemgestaltung steht die eigentliche Problemlösung / Lösungsfindung im Vordergrund. Somit das eigentliche Problem selbst, gedanklicher Lösungsansatz, Abgrenzung zur Umwelt, Anforderungen, die Lösung im engeren inhaltlichen Sinne, Aspekte des Entwurfs, der Architektur des Aufbaus der Lösung, die detaillierte Ausgestaltung etc.

Gesamtheitliche Sicht auf die Projektabwicklung Die beiden Makroprozesse des Projektmanagement und der Systemgestaltung ergeben eine gesamtheitliche Sicht auf die Abwicklung von Projekten:

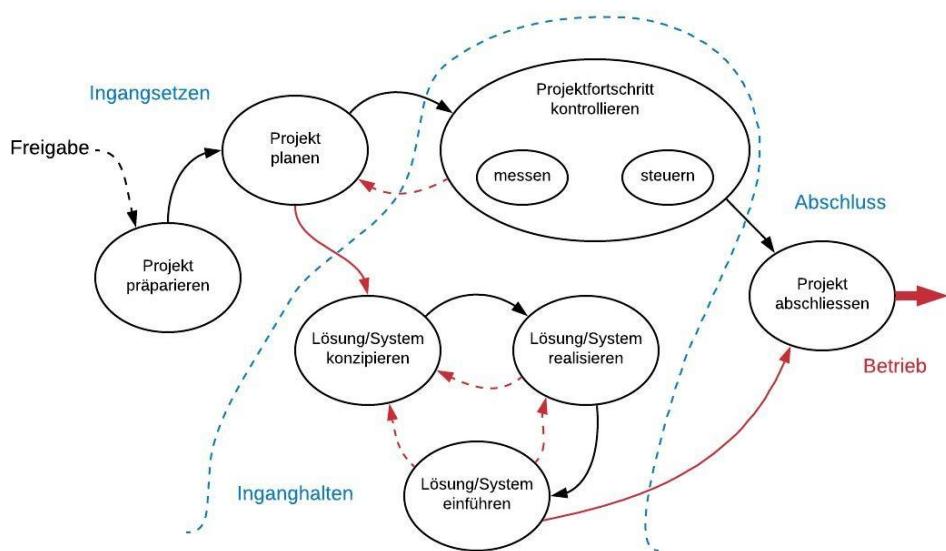


Abbildung 1: Gesamtheitliche Sicht auf Mikroprozesse der Projektabwicklung

Business Case Neuere Denkmodelle beinhalten ebenfalls die Betriebsphase in den Projektüberlegungen. Der effektive Erfolg eines Projekts manifestiert sich erst in langjähriger Anwendung des erhaltenen Systems, der Erfolg eines Projekts kann also erst in der Betriebsphase wirklich überprüft werden. In einigen Projektmethoden wird der Erstellung und ständiger Überprüfung des Business Case hohes Gewicht zugesprochen. Der **Business Case** ist die unternehmerische Rechtfertigung eines Projekts.

Phasenkonzept & Meilensteine Ein Projekt wird wesentlich in Phasen unterteilt, um die Gesamtaufgabe in sinnvolle, besser überschaubare Teile zu gliedern.

Projektphase: zeitlicher Abschnitt eines Projektablaufs, sachlich von anderen Abschnitten getrennt.

Meilenstein: beschreibt Zustand einer Leistung, eines oder mehrere entscheidende Lieferobjekte, die in definierter Qualität bis zu einem bestimmten Zeitpunkt mit entsprechenden Kosten erstellt werden müssen.

Phasen bilden die Schnittstelle zwischen Projektmanagement und Systemgestaltung. Sie erlauben offizielle „Marschhalte“ in der Projektabwicklung. Meilensteine trennen die einzelnen Schritte der Makroprozesse, können aber auch in einzelnen Projektschritten angewendet werden.

1.2 Projektmanagement

1.2.1 Projektmethoden

Eine Projektmethode ist ein auf einem Regelsystem aufbauendes Verfahren zur Erlangung praktischer Ergebnisse. Sie beinhaltet somit auch Themen der Aufbauorganisation, von Rechten und Pflichten (AVK - Aufgaben, Verantwortung und Kompetenzen), von normativen Vorgaben, von funktionalen, institutionellen, instrumentellen und personellen Aspekten. Sie definiert sich in einem kasualen / prozessualen Modell (bspw. Vorgehensmodell) und zusätzlichen Beschreibungen (Rollen, Organisation, Verantwortlichkeiten, Standards etc.) Sie können wiederverwendet werden und dienen als Orientierungshilfen für andere Personen mit ähnlichen Problemen / Projekten.

Herausforderung / Nutzen Die Herausforderung ist, für ein bestimmtes Projekt die geeignete Projektmethode zu eruieren und korrekt anzuwenden. Eine korrekte Projektmethode hilft dabei...

- ... Qualität, Termin- und Kostentreue zu verbessern / einzuhalten
- ... Abwicklungseffizienz von Projekten zu verbessern
- ... unterstützt Beteiligte in Projekten bei der Führung und Durchführung
- ... unterstützt Projektleitung durch Einsatz standartisierter Werkzeuge
- ... Kommunikation innerhalb Projektteam / Unternehmen / externer Stakeholder zu verbessern
- ... gewonnenes Know-How (best practices) wieder anzuwenden, somit von anderen zu profitieren
- ... Projekte untereinander besser zu koordinieren (Priorisierung) und aufeinander abzustimmen (Ressourceneinsatz)
- Abhängigkeiten zu Personen & Firmen zu reduzieren

Typisierung von Projektmethoden Da in einem Projekt überlicherweise ein Produkt / System realisiert wird, sollte man die Typisierung aufgrund von Produkteigenschaften vorzunehmen. Es sind bereits folgende Methoden bekannt:

- Traditionelle Methode - Wasserfall (sequentielle Methoden, plan-driven)
- Agile Methode - Scrum (inkrementelle methoden, event-driven)
- Hybride Methoden - SoDa (Software Development agile)

1.2.2 Grundlegende Denkansätze bei Projektmethoden

Konstruktivistischer (sequenzieller) Ansatz

- Vollständiges Produkt wird in einem Durchlauf zu 100% Umfang (nach Zielen und Anforderungen) realisiert und am Ende des Entwicklungszyklus zur Einführung oder zum Kauf freigegeben
- Ziel / Zielgrösse wird nach deren Festlegung nicht mehr verändert
- Alles zu 100%: Einführung als Big Bang oder parallel zu altem System
- Typischer Ansatz für monolithische Systeme (bspw. Bauwerk (Tunnel), neue SW-Lösung etc.)

Inkrementeller Ansatz

- Eigenständiger Teil des Gesamtprodukts vollständig fertigstellen und freigeben, weitere eigenständige Teile werden parallel / verzögert oder anschliessend fertiggestellt und freigegeben
- 1. Teil zu 100% fertiggestellt, ein 2. Teil ebenfalls zu 100%, n-ter Teil ebenfalls etc.
- Ziel/Ziele (Zielgrösse) werden nach Festlegung nicht mehr verändert
- Typischer Ansatz für modulare Systeme wie Standard-SW mit versch. Modulen, welche unabhängig voneinander realisiert & eingeführt werden können

Evolutionärer Ansatz

- Ganzes Produkt wird initial auf einem tiefen Funktions- / Qualitätslevel fertiggestellt und freigegeben, in weiterer Entwicklungsstufe wird das Produkt funktional / qualitativ weiterentwickelt
- Pro Stufe/Release immer grösserer Funktionsumfang
- Ziel / Ziele (Zielgrösse) kann nach jedem Release verändert werden, muss aber nicht, und dies immer aufbauend auf dem bestehenden Release-Stand
- Typischer Ansatz bei unklaren Aufgabenstellungen

Iterativ / Iterationen *Inkrementell* und *evolutionär* sind iterative Ansätze, da sie die Phasen mehrmals durchlaufen.

Auswahl an Vorgehensmodellen PmBoK - Project Management Book of Knowledge, Hermes - der Eidgenosse, Prince 2, ISO 21500:2012, DIN 69901, Wasserfall, diverse V-Modell(e), Kanban, Scrum, DevOps, SAFe, (fast) Prototyping, eXtreme Programming (XP), Rational Unified Process (RUP), Lean Project Management, 6-Sigma, ICB (IPMA Competence Baselines), firmenspezifische Projektmanagement-Systeme (PMS), ITIL - IT Infrastructure Library (mehr als eine PM), CMMI - Capabilty Maturity Model Integration, System Engineering (SE, mehr als eine PM)

1.3 Systemgestaltung

Die Wissenschaft der Gestaltung von Systemen nennt sich System Engineering (SE). Dazu gibt es ein ganzes Buch, welches aber nicht zum Stoff gehört. Folgendes Denkmodell soll jedoch veranschaulichen, welche Kompetenzbereiche dem System Engineering angehören:

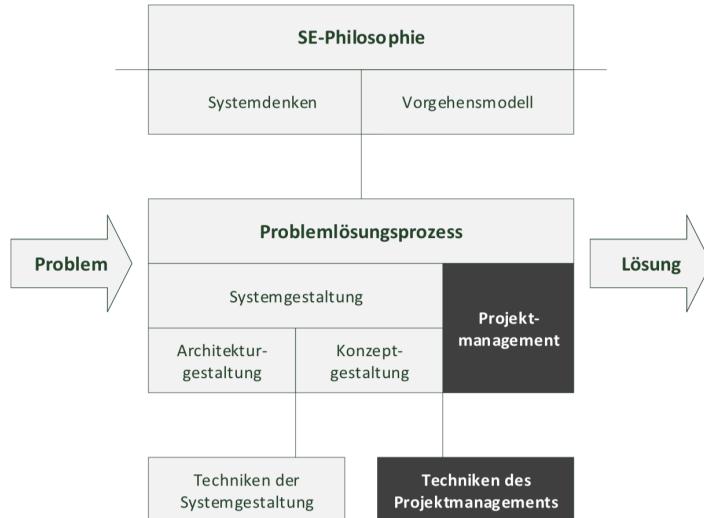


Abbildung 2: Denkmodell zum System Engineering

Projektmanagement in diesem Modell ist Teil des übergeordneten *Problemlösungsprozesses*; anderer Teil ist der gestalterische Aspekt der Projektarbeit (*Systemgestaltung*). Dies ist unterteilt in die Disziplinen *Architekturgestaltung* und *Konzeptgestaltung*. Architekturgestaltung enthält das grundlegende Lösungsprinzip, Konzeptgestaltung die konkrete & umsetzungsorientierte Ausarbeitung der Lösung. Folgt dem Denkansatz „Vom Groben zum Detail“.

Randbedingungen sind eine immer komplexere Welt (vernetzter und dynamischer), gekoppelt mit ungenügenden Informationen darüber.

1.3.1 Systemdenken

Systemdenken wird definiert als eine Denkweise, die es ermöglicht, komplexe Erscheinungen (= Systeme) besser verstehen und gestalten zu können.

1.3.2 Systemmerkmale und -Struktur

Elemente & Beziehungen

- **Systeme** bestehen aus **Elementen (Komponenten)**
- Elemente haben Eigenschaften und Funktionen
 - **Eigenschaften** können Attribute sein (Farbe, Material, Zustand etc.)
 - **Funktionen** geben Zweck eines Elements in konkretem Kontext wieder
- Elemente können als Systeme betrachtet werden, die man wiederum in Elemente gliedern kann
- Elemente sind untereinander durch **Beziehungen** verbunden. Beziehungen können gerichtet sein (wirkungsorientiert). Beziehungen können als Informationsflüsse, Materialflüsse, Lagebeziehungen etc. verstanden werden.
- Elemente können dynamisch sein (zu unterschiedlichen Zeitpunkten eine unterschiedliche Ausprägung haben). Solche dynamischen Systeme werden als komplex betrachtet.

Ordnung Elemente & Beziehungen stellen ein Gefüge / eine Einheit dar und weisen dadurch eine Ordnung auf. Die Ordnung wird als eine Struktur welche *hierarchisch*, *sternförmig* oder *any-to-many* (*Netzstruktur*) sein kann.

1.3.3 Darstellungsarten

Zur Darstellung von Systemen bieten sich folgende Arten an:

- Graphen / Knoten-Kanten-Diagramme, Elemente werden als Knoten (Kreise/Rechtecke), Beziehungen als Kanten (Pfeile, gerichtete/ungerichtete Beziehungen) dargestellt
- Graphen / CLD gemäss Sterman, einensich zur Darstellung der Dynamik von Systemen
- Matrizen, Elemente als Spalten/Zeilen, Beziehungen in jeweiligen Knotenpunkten markieren

1.3.4 Denkansätze und Betrachtungsweisen

Blackbox Innere Struktur eines Systems interessiert in der Betrachtung nicht, nur Inputs und Outputs. Hilfreich, um die Komplexität zu reduzieren.

Whitebox Innere Struktur (Zusammenhang Inputs/Outputs) ist für die Betrachtung von hoher Relevanz. Wenn das System bspw. weiter unterteilt oder simuliert werden soll.

Greybox Zwischenform der Detailierung, System ist grob und teilweise strukturiert.

Aspekte Systeme unter verschiedenen Gesichtspunkten analysieren. Je nach Nutzer interessiert jeden ein anderer Aspekt des gesamten Systems.

1.3.5 Systemtypen

Einfaches System Einfach zu verstehen und beschreiben. Typisiert sich, indem es nur wenige Elemente aufweist, welche durch wenige, nicht-dynamische Beziehungen verbunden sind. Einfache Systeme können in ihrer Gesamtheit explizit beschrieben werden.

Kompliziertes System (massiv vernetzt) Wenn Anzahl und Vielfalt von Komponenten & Beziehungen zunehmen, nennt man es ein massiv vernetztes kompliziertes System. Beziehungen sind jedoch immer noch nicht-dynamisch. Durch Systemgrösse wird es jedoch sehr schwierig, ein solches System explizit zu beschreiben.

→ Technische Systeme

Kompliziertes System (dynamisch) Beziehungen zwischen Elementen verändern sich über die Zeit, sowohl hinsichtlich der Art der Interaktion, Stärke und Struktur. Solche Systeme sind aufgrund dynamischer Beziehungen schwierig quantitativ zu beschreiben bzw. ihr Verhalten vorherzusagen.

→ Sozio-technische Systeme

→ Technische Systeme

Komplexes System Hohe Vielfalt und hohe Dynamik in den Beziehungen zeichnen ein komplexes System aus. Sind sehr schwer, wenn überhaupt, zu beschreiben.

1.3.6 Anwendung des Systemdenkens im Projektmanagement

SEUSAG-Analyse als systemtheoretischer Ansatz zur Bestimmung des Projektgegenstandes (resp. zum Umgang mit Komplexität)

- S** Systemgrenzen bilden
- E** Einflussgrößen ermitteln
- U** Unter- und Teilsysteme abgrenzen
- S** Schnittstellen definieren
- A** Analyse der Unter- und Teilsysteme
- G** Gemeinsamkeiten ermitteln

Agilität von Systemen Man muss mit dem Gedanken spielen können dass es eine Anforderung sein kann, ein agiles Lieferobjekt zu erstellen und nicht nur agil vorzugehen. Dies muss unterschieden werden können und wird im dazugehörigen Dokument vom ILIAS genauer beschrieben. (*agile SYSTEMS Engineering versus AGILA SYSTEMS Engineering*)

1.3.7 Dynamik von Systemen

Policy Resistance Sozio-technische Systeme verhalten sich selten wie vorhergesagt oder erwünscht. Dieses Phänomen der „Policy Resistance“ wird darin begründet, dass Menschen die Feedbacks ihrer Handlungen nicht verstehen und ein rein lineares Denken in Abfolgen praktizieren
 → Event-oriented View

Unsere Aktionen verändern jedoch den Zustand eines Systems, andere Teilnehmer des Systems reagieren um das Gleichgewicht wiederherzustellen. Aktionen haben immer Nebeneffekte zur Folge, welche erkannt werden müssen.

→ Feedback-oriented View

Feedback Loops Die höchste Komplexität resultiert aus den Interaktionen zwischen Komponenten, nicht von den Komponenten selbst. Es werden zwei Arten von Feedback Loops unterschieden:

- Positive Loops (self-reinforcing)
- Negative Loops (self-correctiong, balancing)

Ein System besteht, unabhängig der Komplexität, aus einer Struktur von positiven und negativen Feedbacks. Die Systemdynamik stammt aus den Interaktionen zwischen diesen Feedback Loops. Feedbacklogik wird in Causal Loop Diagrams (CLD) dargestellt.

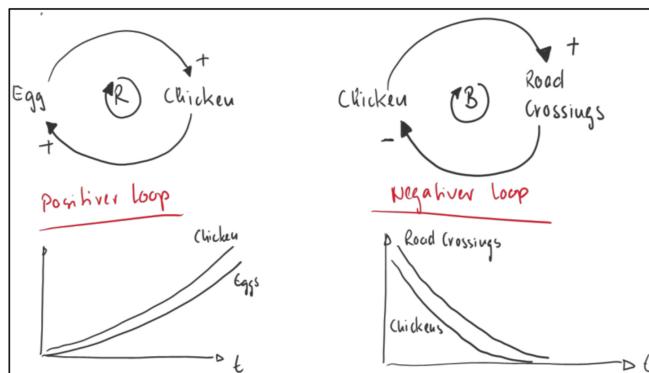


Abbildung 3: Causal Loop Diagram als Beispiel

- + Effekt steht positiv in Beziehung zur Ursache
 (mehr/weniger vom einen hat mehr/weniger vom anderen zur Folge)
- Effekt steht negativ in Beziehung zur Ursache
 (mehr/weniger von einem hat mehr/weniger vom anderen zur Folge)

1.3.8 SE-Vorgehensmodell

Im Software Engineering definiertes Vorgehensmodell, bestehend aus 4 Pfeilern / Komponenten, die in Kombination angewendet werden.

Top-Down - Vom Groben ins Detail

- *Grundidee:*
Von der Blackbox zur Whitebox, Details verhindern oft den Blick auf das Ganze.
- Das Betrachtungsfeld weiter fassen und schrittweise einengen. Betrifft die Untersuchung des Problemfelds, der Ausgangssituation und den Entwurf von Lösungen.
- Bei Untersuchung der Ausgangssituation (Problemfeld) nicht mit detaillierten Erhebungen beginnen, bevor das Problemfeld grob strukturiert, in seine Umwelt eingebettet / gegen sie eingegrenzt ist und Schnittstellen definiert sind (oft auch nur Arbeitshypothese)
- Bei Lösungsgestaltung erst generelle Ziele & Lösungsrahmen festlegen. Detaillierungs- und Konkretisierungsgrad im Verlauf der Projektarbeit schrittweise vertiegen. Konzepte auf höheren Ebenen als Orientierungshilfen für detaillierte Ausgestaltung der Lösung.

Variantenbildung - Denken in Varianten

- *Grundidee:*
Vielfalt von Lösungsmöglichkeiten Rechnung tragen
- Unverzichtbarer Bestandteil guter Planung. Methodische Grundhaltung, muss nicht zu nennenswertem zusätzlichem Arbeitsaufwand führen.
- Nichtbeachtung: grosses Risiko, dass andere Lösungsansätze erst später im Projektverlauf eingebracht werden. Konsequenzen wären Abwürgen der Diskussion, Planungsstopp und Rückkehr auf eine höhere Ebene.

Projektphasen - Makrologik

- *Grundidee:*
Gliedern eines Projekts nach zeitlich-logischen Gesichtspunkten zur Kontrolle von Komplexität
- Nach zeitlichen Gesichtspunkten gegliedertes Raster, hilft den Werdegang einer Lösung in überschaubare Teiletappen zu gliedern.
- Stufenweiser Planungs-, Entscheidungs-, Realisierungsprozess mit vordefinierten Marschhalten / Korrekturpunkten, reduziert Komplexität der Projektabwicklung und schafft Lernchance für Planenden, Durchführenden & Auftraggeber.
- Konzept kann im Sinne einer Makrologik als managementorientierte Komponente gesehen werden. Fordert zwischen Entwicklert & Arbeitgeber/Management an vordefinierten Stellen eine Kontaktnahme, Willensbildung und Entscheidung.

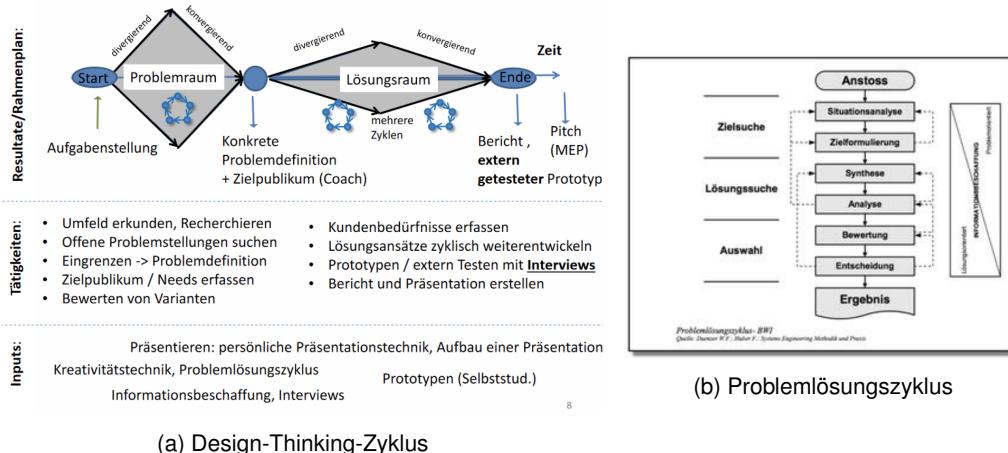
Problemlösungszyklus - Mikrologik

- *Grundidee:*
Systematisches Vorgehen bei Auftreten jeder Art von Problemen in jeder Phase des Projekts
- Als eine Art Mikrologik ein Leitfaden zur Behandlung von Problemen oder Aufgabestellungen, in jeder Phase eines Projekts anwendbar.

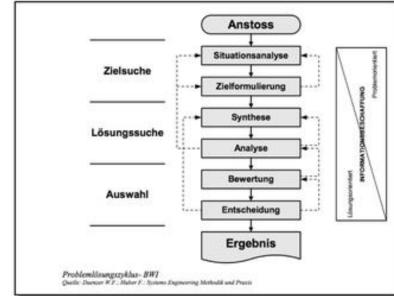
1.3.9 Problemlösungszyklus - PLZ

- **Problem:** Differenz zwischen „Ist-“ und „Soll-Zustand“
- Versuchs-Irrtum-Methode (try and error): natürliche Art und Weise, Mensch kann jedoch den Try & Error Prozess vorwegnehmen, bevor er eine Entscheidung treffen muss. Er besitzt die Fähigkeit, Folgen seines Handelns abzuschätzen. Dies erlaubt ihm systematische Vorgehen zur Problemlösung.

Design Thinking Möglichkeit zur systematischen Vorgehensweise zum Problemlösen.



(a) Design-Thinking-Zyklus



(b) Problemlösungszyklus

Der **Anstoß** setzt den PLZ in Gang. Er startet entweder das ganze Projekt oder befindet sich innerhalb einer Phase oder eines Arbeitspakets als Auslöser des PLZ.

Problemlösungszyklus - Zielsuche

- Teilschritte der Zielsuche sind die *Situationsanalyse* und *Zielformulierung*
- Zweck der **Situationsanalyse**:
 - Erarbeitung grundlegendes Verständnis bzgl. des Problems, dessen Erscheinungsformen sowie Ursachen und deren Zusammenhänge
 - Strukturierung & Abgrenzung des Problems bzw. Untersuchungsfelds
 - Definition des Eingriffs- und Gestaltungsbereichs für Lösungssuche
 - Schaffung der notwendigen Informationsbasis für nachfolgende Arbeitsschritte
- Zweck der **Zielformulierung**:
 - Alle Ziele, die zur Lösung eines Problems erreicht werden, müssen möglichst *vollständig, präzise und verständlich, realistisch* und *lösungsneutral* definiert werden
 - Unterscheiden zwischen Muss-, Soll- und Wunschzielen
→ Diese Unterscheidung setzt Prioritäten der Ziele in der Projektabwicklung
 - Zielvorstellung(-en) können bereits durch *Anstoß* vorgegeben sein
 - Durch diese beiden Teilschritte sollen Randbedingungen einer Problemlösung definiert werden, es darf keineswegs ein bestimmter Lösungsweg vorgegeben werden
(→ Lösungsneutralität der Ziele)
- Methoden/Techniken/Werkzeuge:
 - **Informationsbeschaffung:** Befragung, Beobachtung, Checklisten, Fragebogen, Interview, Panelbefragung, Prognosen, Stichproben etc.
 - **Informationsaufbereitung:** ABC-Analyse, Ablaufdiagramm, Analysetechniken, Benchmarks, Korrelationsanalyse, Math-Statistik, Regressionsanalyse, Schwachstellenanalyse etc.
 - **Informationsdarstellung:** Darstellungstechniken, Mind Maps, Input-Output-Methode etc.
 - **Zielformulierung:** Operationalisierung, Target Costing, Use Case etc.

Problemlösungszyklus - Zielsuche

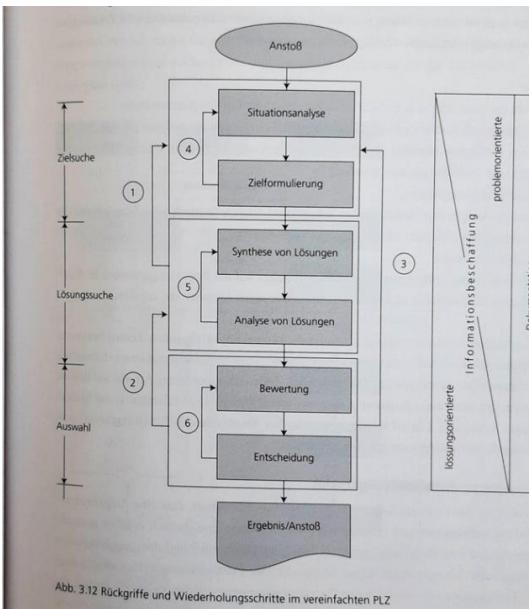
- Teilschritte der Lösungssuche sind **Synthese** und **Analyse** von Lösungen
- Zweck der **Synthese**:
 - Sammlung & Ausarbeitung von Lösungsalternativen.
 - Kreativer, aufbauend-konstruktiver Schritt, baut auf Resultaten der Situationsanalyse auf (Situationskenntnis, Problemverständnis, Zielformulierungen, Anforderungskatalog)
- Zweck der **Analyse**:
 - Machbarkeit einer Lösungsalternative verifizieren
(ohne zu werten oder mit anderen Alternativen zu vergleichen, erfolgt in *Auswahl*)
 - Kritischer, analytisch-destruktiver Schritt
 - Erfüllt eine Lösungsalternative die Ansprüche nicht, kann sie verworfen werden oder muss überarbeitet werden
- Methoden/Techniken/Werkzeuge:
 - **Synthese - Kreativität**: Analogiemethode, Bionik, Brainstorming, Just in Time, Kärtchentechnik, Kreativitätstechniken, Morphologie etc.
 - **Synthese - Optimierung**: Operations Research, Dynamische Optimierung, Entscheidungsbaum, Lineare Optimierung, Spieltheorie, Warteschlangenmodelle, Use Case etc.
 - **Analyse**: FMEA, Fehlerbaumanalyse, Gefährdungsanalyse, Reverse Engineering, Risk Management, Schwachstellenanalyse, Total Quality Control, Wert-Analyse etc.

Problemlösungszyklus - Zielsuche

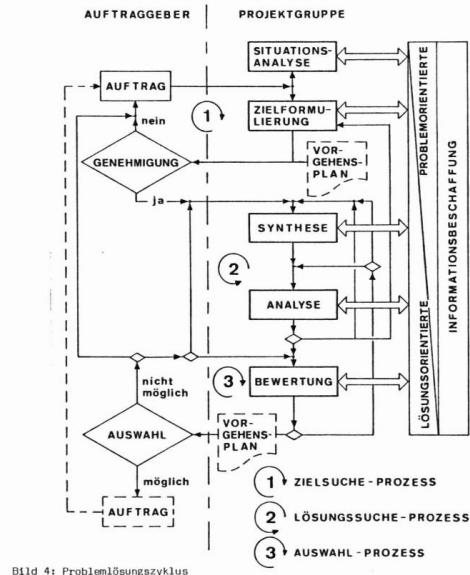
- Teilschritte der Auswahl sind **Bewertung** und **Entscheid**
- Zweck der **Bewertung**:
 - Taugliche Lösungsvarianten gegenüberstellen (jene, welche alle Muss-Ziele erfüllen)
 - Formale Bewertung wird angezeigt, wenn keine Variante als Beste identifiziert werden kann
 - 3 Bedingungen müssen erfüllt werden:
 - * Min. 2 echte Lösungsalternativen müssen existieren
 - * Bewertungskriterien (bezogen auf Zielformulierungen) müssen definiert sein
 - * Bewerter müssen Alternativen hinsichtlich Bewertungskriterien beurteilen können
- Zweck des **Entscheids/Auswahl**:
 - Basierend auf Bewertungsergebnissen die weiter zu bearbeitende Variante bestimmen
- Methoden/Techniken/Werkzeuge:
 - **Bewertung & Entscheidung**: Analytic-Hierarchie-Process, Bewertungstechniken, Entscheidungstheoriebaumverfahren, Kosten-Nutzen-Analyse, Kriterienplan, Nutzwerkanalyse, Wirtschaftlichkeitsrechnung, Punktbewertung etc.

Es kann sein dass PLZ keine zufriedenstellende Variante hervorbringt. Oder dass sich das Projekt mit verfügbaren Mitteln nicht realisieren lässt. Es sind Schlussfolgerungen möglich:

- Projektabbruch
- Sistierung der Zusammenarbeit mit Systementwicklern
- Zielreduktion
- Zurück auf höhere Systemebene, mit anderen Wegen zum Ziel kommen
- Problem neu umschreiben / definieren



(a) Aufteilung des PLZ in Grob-/Feinzyklen



(b) Erweiterter PLZ

Erweiterter PLZ Im Erweiterten PLZ werden die Schritte bezogen auf Rollen Auftraggeber und Projektgruppe unterschieden. Unter Projektgruppe ist auch der Projektmanager miteinbezogen, welcher den PLZ begleitet und führt.

1.4 Erfolgsmuster in der Projektarbeit

Erfolg / Misserfolg?

- **Wer entscheidet darüber?**
Auftraggeber, Stakeholder, Endbenutzer des neuen Systems?
- **Wann wird darüber entschieden?**
Nach Projektende, nach Teil der Nutzungsphase?
- **Zusätzliche Themen:** Standish Chaos Report, PMI (Dokumente dazu im ILIAS)

2 Project Management - Projekt in Gang setzen

2.1 Projekt präparieren

Projektorganisation Massgebende Entscheidungsgremien initialisieren:

- Reine Projektorganisation / Linien-PO
- Stab-Linien-Projektorganisation / Projektkoordination
- Matrix-Projektorganisation
- Mischformen

(Erste) Ziele setzen Ziele sollen Lösungssuche steuern und eine Lösung nicht nachträglich rechtfertigen. Erarbeitung der Zielsetzungen ist Tätigkeit des Projektmanagements. Zielsuche ist die erste Phase des PLZ, Zielformulierung der 2. Schritt.

2.2 Zielbegriff

Definitionen

- Ziel: angestrebter Zustand / erwünschte Wirkung
- Zukünftige Ergebnisse, die durch Massnahmen / Lösungen erreicht werden sollen
- Ziele sollen bekannt sein, bevor Ermittlung der Anforderungen begonnen wird
- Jede Anforderung muss sich auf ein Ziel zurückführen lassen
- Unterscheiden zwischen:
 - **Abwicklungsziele**
Unterteilt in Leistungs-, Qualitäts-, Zeit/Termin- und Kostenziele
 - **Systemziele**
Fokussieren auf Akzeptanz & Wirtschaftlichkeit des zu erstellenden Systems. Hierarchisch unterteilt in Global-, Gruppen- und Detailziele
- Ziele sollen folgenden Prinzipien folgen:
 - Wertorientierung
 - Lösungsneutralität (keine mögliche Lösung favorisieren / vorgeben)
 - Operationalität (Ziel-Mittel-Denken)
 - Vollständigkeit der Zielinhalte
 - Berücksichtigung der wichtigen Informationsquellen & Interessenslagen
 - Feststellbarkeit der Zielerfüllung
 - Prioritätensetzung
Muss-Ziele müssen zwingend eingehalten werden
Soll-Ziele hohe Bedeutung, wenn möglich einhalten
Wunschziele tiefe Verbindlichkeit, „nice to have“
 - Widerspruchsfreiheit, Zielkonflikte vermeiden / eliminieren
 - Überblickbarkeit & Bewältigbarkeit des Zielkatalogs

Zielformen

Abwicklungsziele Bezug auf Projektmanagement, geben vor was bei Meilensteinen konkret vorliegen muss (Lieferobjekte, Meilensteinplan)

Vorgehensziele Bezug auf Projektmanagement, geben vor wie das Projekt abgewickelt wird, bspw. systematisches Stakeholder-Management durchführen

Systemziele Bezug auf Systemgestaltung, Teil des Problemlösungsprozess. Sagen aus was mit zu gestaltender Lösung erreicht / vermieden werden soll (Zustand).

2.3 Anforderungsbegriff

- Anforderung: eine Aussage über eine Eigenschaft oder Leistung eines Produkts, Prozess oder der am Prozess beteiligten Personen

- Unterscheidung:

Ziel Zustand

Anforderung Eigenschaft (Verhalten, Fähigkeit, Funktion)

2.4 Ziel-Mittel-Denken

- Kann ein Ziel („WAS“) zu einem Mittel („WIE“) werden?
- Ziel-Mittel-Denken: Instrument zur Entwicklung und Operationalisierung von Zielen
- Nutzt zur Visualisierung eine Ziel-Mittel-Hierarchie
- Prinzip der Lösungsneutralität muss jederzeit sichergestellt sein!

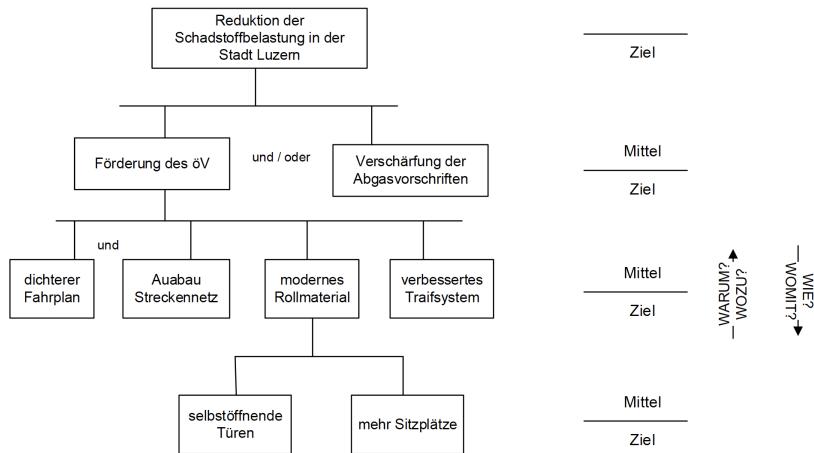


Abbildung 6: Ziel-Mittel-Denken

S.M.A.R.T. Objectives

- Ziele sollten möglichst SMART sein:
- S** Spezifisch
Grenzen des Zielbereichs werden vorgegeben, für konkrete Organisation o.ä.
- M** Messbar
Qualitäten / Quantitäten der Zielerreichung bestimmbar, Indikatoren ableitbar
- A** Angemessen
Das „richtige“ Ziel? Besteht Bedarf für geplante Massnahme?
- R** Realistisch
Aussichten, Ziel zu erreichen, unter Rahmenbedingungen ausreichend hoch
- T** Terminiert
Zeitlicher Rahmen wird vorgegeben
- Ziele müssen nicht auf jeder Stufe s.m.a.r.t. formuliert sein
 - Zur Messbarkeit der Zielerreichung: Prinzip der Feststellbarkeit der Zielerfüllung (Kap. 2.2)
→ setzt s.m.a.r.t.e Formulierung voraus

2.5 Zielrelationen (-Matrix)

Ziele können in unterschiedlicher Beziehung zueinander stehen. Ziele sollen dem Prinzip der Widerspruchsfreiheit gehorchen. Dazu werden alle aufgelisteten Ziele auf ihre Relationen überprüft.

- sich gegenseitig unterstützen (**unterstützend**)
 - Erreichung von Ziel A unterstützt Erreichung von Ziel B und umgekehrt
 - **angenehm**
- unabhängig voneinander sein (**indifferent**)
 - Erreichung Ziel A ist unabhängig von Erreichung Ziel B und umgekehrt
 - **problemlos**
- **gegenläufig** sein (Gegenläufigkeit / Zielkonkurrenz)
 - Ziel A & B behindern sich: Je besser A erreicht wird, umso schlechter kann B realisiert werden und umgekehrt
 - **Kompromisse müssen gefunden werden**
- Im Konflikt zueinander stehen (**widersprüchlich**, Zielkonflikt)
 - Ziel A und B logisch oder situationsbedingt im Konflikt, können nicht gleichzeitig nebeneinander existieren
 - **Konflikt muss aufgelöst werden**
 - * Prioritäten setzen (Muss, Soll, Wunsch)
 - * Auflösung des Konflikts durch Innovation
 - * Einführung von Mindest- und Höchstwerten
 - * Streichen eines Konfliktverursachters

2.6 Zielgewichtung

Nicht alle Ziele sind gleich wichtig, sie müssen gewichtetet werden.

2.6.1 Paarvergleich

Methode zur Bestimmung und Berechnung der Präferenzen / Gewichtung von Zielen oder Kriterien untereinander. Funktionsweise: Jedes Ziel wird mit jedem anderen 1:1 verglichen und ein „Sieger“ bestimmt.

$$\text{Mögliche Kombinationen im Paarvergleich: } K = \frac{n(n-1)}{2}$$

2.6.2 Präferenzmatrix

Instrument zum Paarvergleich und der Berechnung der Gewichtung. Jede Zielkombination wird mit dem Sieger (a ist wichtiger als b) benennt. 2 Möglichkeiten zur Berechnung des Gewichts:

$$\text{Gewicht}[\%] = \frac{\text{AnzahlNennungen} \cdot 100}{\text{TotalNennungen}}$$

$$\text{Gewicht}[\%] = \frac{\text{umgedrehter Rang} \cdot 100}{\text{SummeAllerRaenge}}$$

(2. Variante hat den Vorteil, dass Ziele ohne Nennung ein Gewicht erhalten und nicht 0% Gewichtung haben)

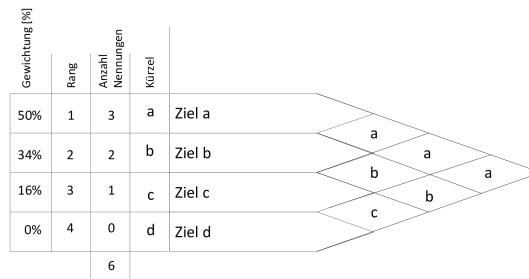


Abbildung 7: Beispiel Gewichtung in Präferenzmatrix

2.6.3 Analytical Hierarchy Process (AHP)

Alternative Methode zur Berechnung der Zielgewichte. Man kann hiermit bestimmen, um wieviel ein Ziel wichtiger ist als ein anderes. Zur Berechnung wird jedoch Vektorgeometrie benötigt, was wir dann gar nicht angeschaut haben.

2.6.4 Lieferobjekte aus „Projekt präparieren“

Aus diesem ersten Schritt resultiert ein Projektantrag, welches als Vertrag für die Aufnahme der aufwändigen Planungsaktivitäten dient.

Ziel → Projektantrag liegt im Status „approved“ vor
Lieferobjekt → Projektantrag, freigegeben

2.7 Projekt planen (Planungsprozess)

Im Planungsprozess werden Pläne (= dokumentarische Lieferobjekte) entwickelt. Sie beinhalten Planung, sind gleichzeitig aber auch immer die erste Analyse der Aufgabenstellung. Der Problemlösungszyklus PLZ wird ein erstes Mal durchlaufen. Er ist in 9 Schritte aufgeteilt:

- **Meilensteinplanung** → Abwicklungszielplan (Meilensteine)
- **Projektstrukturplanung** → Projektstrukturplan (Arbeitspakete & deren Strukturierung)
- **Ablaufplanung** → Ablaufplan (Abhängigkeiten, Projektdauer, Pufferzeiten)
- **Ressourcenplanung** → Ressourcenplan (Durchlaufzeit (Aufwand) der Arbeitspakete)
- **Organisationsplanung** → Organisationsplan (Mitarbeitende zugeteilt auf Arbeitspakete)
- **Kostenplanung** → Projektkostenplan (Dokumentierte Kosten & Finanzierung)
- **Terminplanung** → Terminplan (konkrete Termin- und Einsatzplanung)
- **Budgetplanung** → Projektbudgetplan (kalenderische Finanzierung)
- **Informationsplanung** → Informations-/Dokumentationsplan (Kommunikations-/Infokonzept)

2.7.1 Meilenstein- / Phasenplanung

Input Zielsetzungen an das Projekt (aus Projektantrag)

Ziel Bündelung der Ziele & Umsetzung in definierte Zwischenresultate (Meilensteine)

Kriterien Meilenstein: Zeit, Qualität, Lieferobjekt, Aufwand/Kosten

Output Definierte Meilensteine

Lieferobjekt Abwicklungszielplan

Phasenplan Phasenplan in Anlehnung an gewählte Projektmethode erstellen. Es gibt mehrere Möglichkeiten, wie bspw. nur zwei Phasen oder Phasen mit Benennung (Initiiieren - Konzipieren - Realisieren - Einführen). Man muss nicht zwingend weiter unterteilen, wenn dies nicht nötig ist.

Meilensteinplan

Meilenstein beschreibt einen Zustand einer Leistung (ein oder mehrere Lieferobjekte), die bis zu einem bestimmten Zeitpunkt in der definierten Qualität zu den entsprechenden Kosten erstellt werden müssen.

Meilensteine normalerweise zwischen den Phasen des Phasenplans definieren. Es können auch innerhalb von Phasen Meilensteine vorgesehen werden, wenn dies der Absicherung von Risiken oder Qualität sinnvoll erscheint. Bei einem Meilenstein wird das Projekt „eingefroren“. Projekt wird je nach Lieferobjekt in die nächste Phase freigegeben oder Korrekturen an Lieferobjekten werden vorgenommen. Meilenstein kann eine „Exit“-Möglichkeit sein, um das Projekt zu stoppen.

2.7.2 Projektstrukturplanung (PSP)

Input Meilensteinplan

Ziel Gliederung des gesamten Projekts in plan- und kontrollierbare Arbeitspakete

Output Definierte Arbeitspakete & deren Strukturierung

Lieferobjekt Projektstrukturplan (PSP)

Arbeitspaket

Ein Arbeitspaket ist eine in sich geschlossene Aufgabenstellung innerhalb eines Projekts. Kann bis zu einem festgelegten Zeitpunkt mit definiertem Ergebnis und Aufwand vollbracht werden.

- Arbeitspaket ist die tiefste Gliederungsebene des PSP, somit auch die kleinste Planungseinheit
- Sie werden einem Projektmitarbeiter oder Projektteam zugewiesen
- Faustregeln für den Umfang eines Arbeitspaketes:
 - Zeit- und Kostenanteil pro AP: 2-5% des Gesamtprojekts
 - Dauer in Abhängigkeit des Berichtsintervalls: Faktor 1.5
(bspw. bei Berichtsintervall von 4 Wochen ist Dauer des AP max. 6 Wochen)
- Arbeitspaket soll auch ein „messbares“ deliverable beinhalten (analog Meilenstein), da zwischen APs Abhängigkeiten bestehen
- Nachläufer (nachfolgendes AP) kann nur gestartet werden, wenn Vorläufer (vorhergehendes AP) korrekt abgeschlossen wurde
- Hierarchische Struktur:
 - Programm → 2-n Projekte → 2-n Teilprojekte → 2-n Aktivitäten (mündl. Delegation an MA)

PSP - Objekt- oder vorgehensorientiert

- **Work-Breakdown-Structure (WBS)**

Vorgehensorientierter PSP strukturiert die APs entlang der Phasen, in ihrer logischen zeitlichen Folge. Kann als wiederverwertbarer Standardrahmen für die Abwicklung von Projekten in einem Unternehmen genutzt werden.

- **Product-Breakdown-Structure (PBS)**

Objektorientierter PSP (Produktstrukturplan) definiert APs entlang des zu erstellenden Produktes/Systems. Einmaliges Werk, kann meistens nicht wiederverwendet werden.

PSP - Wertorientiert Wertorientierter PSP macht Sinn, falls der „Wert“ eines APs auf den Wert des gesamten Projekts appliziert werden soll. Kann sinnvoll sein, wenn bspw. ein AP aus Kostengründen weggelassen werden soll oder nicht.

Formale Regeln im PSP AP ist die kleinste Planungseinheit. Für Schätzung von Aufwänden in weitere Aktivitäten unterteilbar, muss aber immer wieder zu einem AP zusammengefügt werden.

- AP / Aktivität besteht aus Substantiv + Verb: *Pflichtenheft erstellen*
- Meilenstein als Zustand (Ereignis) zu formulieren: *Pflichtenheft erstellt*

Arbeitspaketliste Resultat dieses Planungsschritts ist eine Arbeitspaketliste, in welcher alle einzelnen Arbeitspakete erfasst werden.

2.7.3 Ablaufplanung

Input Projektstrukturplan (PSP)

Ziel Eruierung von sachlogischen Abhängigkeiten zw. APs & Durchlaufzeiten von APs

Output sachlogische Abhängigkeiten, Projektdauer, Pufferzeiten

Lieferobjekt Ablaufplan

Ermittelt für im PSP identifizierte APs die sachlogischen Abhängigkeiten. Dazu werden Vorläufer / Nachläufer jedes APs bestimmt. Genaue kalendarische Termine sind noch irrelevant, jedoch wird pro AP erste Durchlaufzeit/Dauer geschätzt (aus Erfahrungswerten oder ersten Vorstellungen des PL/Auftraggebers)

Arbeitspaketliste - erweitert Für diesen Planungsschritt wird die AP-Liste um den Block „Vorgänge“ und evtl. auch schon um „Einsatzmittel“ erweitert. Diese beinhalten jeweils folgende Spalten:

- **Vorgänge**
 - Vorgangsdauer (Schätzung der Dauer)
 - Direkter Vorläufer
 - Direkter Nachläufer
- **Einsatzmittelbedarf**
 - Personal
 - Sachmittel
 - Finanzmittel

Schätzung der Dauer von APs Dauer, Aufwand und Personaleinsatz noch unbekannt, APs können mit einer fixen Dauer versehen werden. Er werden erste Schätzungen der Dauer von APs gemacht für einen ersten Anhaltspunkt der Durchlaufzeit des Projekts. Dauer wird so ermittelt, dass ein gewünschter Endtermin des Projekts zustande kommt (ohne Berücksichtigung der effektiven Aufwände & Ressourcen). Kann unter *Vorgangsdauer* in der AP-Liste eingetragen werden.

Sachlogische Abhängigkeit Setzt ein Verständnis der Problemstellung voraus. Abhängigkeiten werden über die Spalten *Direkter Vorläufer* & *Direkter Nachläufer* dargestellt. Eine grafische Darstellung als Balkendiagramm ist möglich.

Netzplanverfahren Ermöglichen Simulationen („was passiert wenn...“) und ein klares Bild der Pufferzeiten wird errechenbar. Es existieren mehrere Arten von Netzplanverfahren.

Netzplanverfahren	Darstellung	Beispiel
Vorgangspfeilnetz		Critical Path Method
Vorgangsknotennetz		Precedence Diagramming Method
Ereignisknotennetz		Program Evaluation and review Technique

Abbildung 8: Übersicht der verschiedenen Netzplanverfahren

Vorgangspfeil-Netzplan (VPN) - Critical Path Methode (CPM)

- Ziel ist im Netzplan zu erkennen, welche Aktivitäten mit und welche ohne Zeitreserve (kritischer Pfad) durchführbar sind (und somit Ausführungsdauer für das Projekt direkt beeinflussen)
- CPM als Planmethode: Darstellungsmittel für Teminplanung & Algorithmus für die Berechnung des kritischen Pfades sowie der Projekt-Gesamtdauer
- Netzplan in CPM-Darstellung mit Knoten und Kanten
- Ermittlung der Puffer: Aus den unterschiedlichen Zeitpunkten berechnen

Darstellungsregeln

- **Knoten:** Darstellungselement für ein Ereignis (Zustand) innerhalb eines Projekts
- **Pfeile/Kanten:** Darstellungselement für einen Vorgang (Tätigkeit, AP) in einem Projekt
(Ein Vorgang entspricht einem AP)

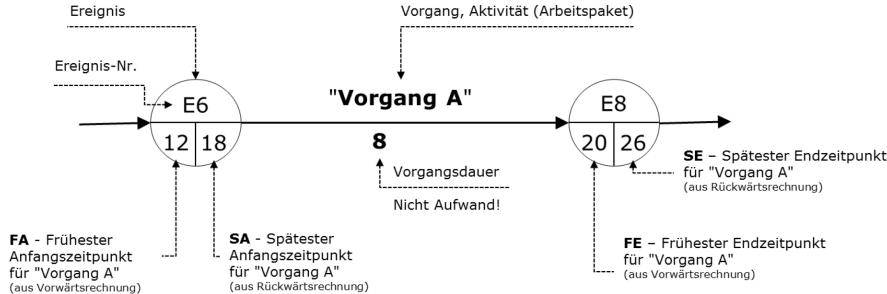


Abbildung 9: VPN-Notation von Knoten & Kanten

Regeln zur Erstellung eines Netzplans nach CPM

- Regel 1** Ein Vorgang kann erst beginnen, wenn alle Vorgänger abgeschlossen sind. Es fällt das Anfangsereignis mit dem Endereignis des vorhergehenden Vorgangs zusammen (Ausnahme erster Vorgang)
- Regel 2** Müssen mehrere Vorgänge beendet sein, bevor ein Nachfolger beginnen kann, so enden sie alle im Anfangsbereich des Nachfolgers.
- Regel 3** Können mehrere Nachfolger beginnen, nachdem ein Vorgänger beendet ist, so beginnen sie alle im Endereignis des Vorgängers.
- Regel 4** Haben zwei oder mehr Vorgänge gemeinsame Anfangs- und Endereignisse, so ist ihre eindeutige Kennzeichnung durch Einfügen von **Scheinvorgängen** (fiktiver Vorgang mit der Dauer 0) sicherzustellen
- Regel 5** Beginnen und enden in einem Ereignis mehrere Vorgänge, die nicht alle voneinander abhängen, muss die Eindeutigkeit ebenfalls durch **Scheinvorgänge** erreicht werden (bspw. Zusammenführung an Meilensteinen)
- Regel 6** In einer Folge von Vorgängen können beliebig viele **Scheinvorgänge** eingefügt werden → logische Verknüpfung und bessere Übersicht, so wenig wie möglich einsetzen
- Regel 7** Kann ein Vorgang beginnen, bevor der Vorgänger vollständig abgeschlossen ist, muss der Vorgänger unterteilt werden
- Regel 8** Jeder Vorgang darf nur einmal ablaufen, es dürfen keine Schleifen auftreten

Vorgang	Dauer	Vorläufer	Nachläufer
A	25		B, C
B	30	A	E
C	10	A	D
D	40	C	F
E	60	B	G
F	5	D	G
G	6	E, F	

Abbildung 10: Vorgangsliste

Beispiel eines Netzplans nach CPM In Vorgangsliste sind pro AP direkte Vorläufer & Nachläufer aufgelistet. Zur Erstellung des Vorgangsdiagramm würde eines davon ausreichen. Daraus wird nun das Vorgangsdiagramm erstellt. Bei diesem fehlen jedoch noch die Zeitpunkte. Die Dauer der Vorgänge werden auf den Pfeilen eingetragen.

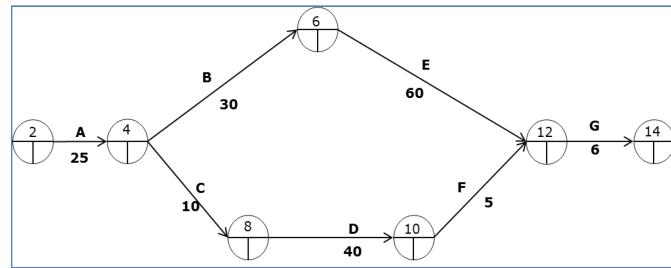


Abbildung 11: Vorgangspfeil-Netzplan ohne Zeitpunkte

Als Nächstes werden die **frühesten Zeitpunkte** ermittelt.

(Werden 2 oder mehr Pfade zusammengeführt, wird als FA des nachlaufenden Vorgangs der höchste errechnete Wert eingefügt, da die frühest möglichen Zeitpunkte gesucht sind)

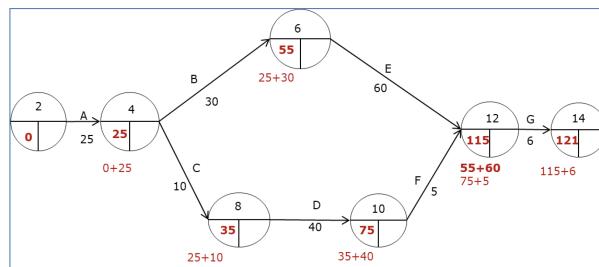


Abbildung 12: VPN: Berechnung der frühesten Zeitpunkte

Als Nächstes werden noch die **spätesten Zeitpunkte** berechnet.

(Werden 2 oder mehr Pfade zusammengeführt, wird als SE des nachlaufenden Vorgangs der tiefste errechnete Wert eingefügt, da die spätest möglichen Zeitpunkte gesucht sind)

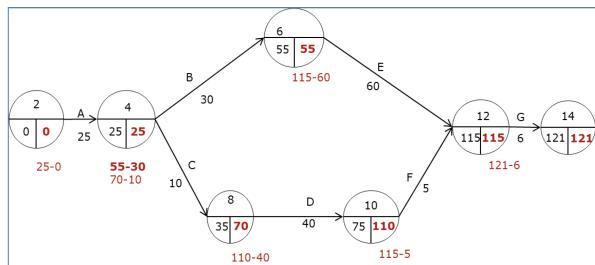


Abbildung 13: VPN: Berechnung der spätesten Zeitpunkte

Nun wird noch der kritische Pfad ermittelt. Dieser besteht aus den Vorgängen, welche keine (gesamte) Pufferzeit aufweisen (in diesem Fall A, B, E, G)

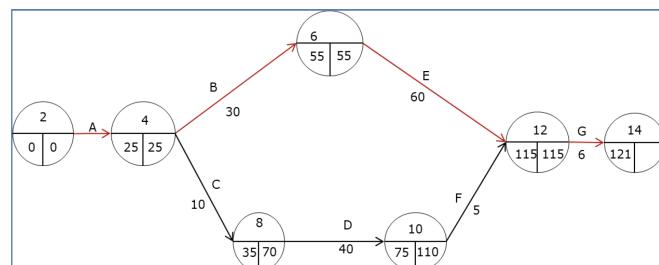


Abbildung 14: VPN: Ermitteln des kritischen Pfads

Precedence Diagramming Method (PDM) Findet sich in toolgestützten PM-Werkzeugen. Pufferzeiten werden ebenfalls durch Vorwärts- / Rückwärtsrechnung errechnet.

- FAZ bis FEZ: Puffer
- SAZ bis SEZ: Dauer des Vorgangs

Pufferzeiten Pufferzeiten geben Auskunft, wie viel Reserve im Ablaufplan beinhaltet ist. Kritischer Pfad gibt Auskunft in welchen APs keine Reserven vorhanden sind bzw. welche Vorgangskette optimiert werden muss, um den Projektendzeitpunkt zu beeinflussen.

(Aufwände & Ressourcen/Skills in diesem Planungsstadium noch nicht berücksichtigt. Diese haben eine entscheidende Auswirkung auf die Dauer von APs)

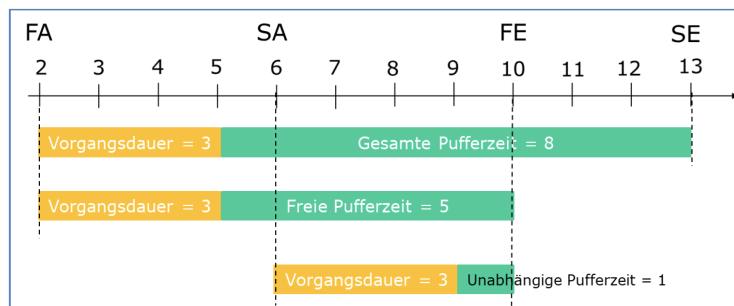


Abbildung 15: Pufferzeiten nach PDM

Berechnung von Pufferzeiten

Gesamte Pufferzeit (GP)

$$GP = SE(B) - FA(A) - Vorgearsdauer$$

Gibt an, wie lange ein Vorgang bzw. sein Anfangszeitpunkt maximal ausgedehnt/hinausgezögert werden darf, ohne dass das nachfolgende AP (und Endzeitpunkt des Projekt) beeinträchtigt wird
Vorgang(-skette) ist **kritisch** (auf kritischem Pfad), wenn GP = 0 ist!

Freie Pufferzeit (FP)

$$FP = FE(B) - FA(A) - Vorgearsdauer$$

Gibt an, wie lange ein Vorgang bzw. sein Anfangszeitpunkt maximal ausgedehnt/hinausgezögert werden darf, ohne dass der früheste Anfangszeitpunkt des nachfolgenden Vorgangs beeinträchtigt wird

Unabhängige Pufferzeit (UP)

$$UP = FE(B) - SA(A) - Vorgearsdauer$$

Gibt an, wie lange ein Vorgang bzw. sein Anfangszeitpunkt maximal ausgedehnt/hinausgezögert werden darf, ohne dass der früheste Anfangszeitpunkt des nachfolgenden Vorgangs beeinträchtigt wird - **unabhängig** darum, weil nicht relevant ist, wann das Anfangsereignis tatsächlich ausgelöst wird

Resultat - Ablaufplan Als Resultat dieses Planungsschrittes entsteht ein Ablaufplan, wobei bei der AP-Liste die Spalten von *Vorgänge* (also *Vorgearsdauer*, *Direkter Vorläufer* und *direkter Nachläufer*) ausgefüllt werden.

2.7.4 Ressourcenplanung

Input Ablaufplan

Ziel Kapazitäten SOLL vs. Kapazitäten IST, Aufwände & Dauer ermitteln

Output Durchlaufzeit/Dauer (Aufwand) der APs

Lieferobjekt Ressourcenplan, Kapazitätsbelastungsdiagramm

FTE - Full Time Equivalent

- Einplanung von personenneutralen Ressourcen (keine konkreten Mitarbeitenden)
- **1 FTE** entspricht einer Vollzeitstelle (Ein Mitarbeiter, der zu 100% im Projekt tätig ist)
(bspw. 1 FTE = 3 Mitarbeitende (1x 50%, 2x je 25%))

Dauer / Aufwand / FTE

Verhältnis von **FTE** zu Dauer & Aufwand:

$$(1) FTE = \frac{\text{Aufwand}[Tagen]}{\text{Dauer}[Tagen]}$$

$$(2) \text{Aufwand}[Tagen] = \text{Dauer}[Tagen] \times FTE$$

$$(3) \text{Dauer}[Tagen] = \frac{\text{Aufwand}[Tagen]}{FTE}$$

• Terminfixierte Planung

Bei personeller Ressourcenplanung ist Dauer eines AP grundsätzlich gesetzt, es sind die benötigten Ressourcen/ Personalkapazitäten (in FTE) pro AP in Abhängigkeit der Dauer & des Aufwands gesucht. Dazu wird **Formel (1)** verwendet (Aufwand muss noch geschätzt werden)

• Ressourcenfixierte Planung (Einsatz verfügbarer personeller Ressourcen)

Sind die personellen Ressourcen fix vorgegeben und der Aufwand ist klar, wird die Dauer des AP änderbar. Man verwendet **Formel (3)**.

Aufwandschätzung / Schätztechniken Schätzung der Aufwände sehr schwierig, Dauer eines AP oder Personaleinsatz in AP hängen davon ab. Aufwände bestimmen letztendlich die Durchlaufzeit & Kosten eines AP und somit des Projekts. Aufwandschätzungen werden periodisch überprüft und damit im Laufe des Projektfortschritts immer genauer.

Parkinson's erstes Gesetz:

Work expands so as to fill the time available for its completion

Arbeit dehnt sich aus, soweit es geht

Aus der Praxis bekannt, ist man zu früh fertig, findet man immer Arbeit zur „Verschönerung“ oder testet mehr etc. → Time Boxing

Techniken zur Aufwandschätzung

- Standard Delphi-Verfahren
- Breitband Delphi-Verfahren
- Beta-Verfahren
- Function Point (→ SW-Development)
- COCOMO (Constructive Cost Model); in vielen Varianten
- LOC (Lines of Code → SW-Development)
- etc.

Kapazitätsbelastung (-sdiagramm) Nachfolgende Abbildung zeigt die berechneten Ressourcen pro Arbeitspaket auf.

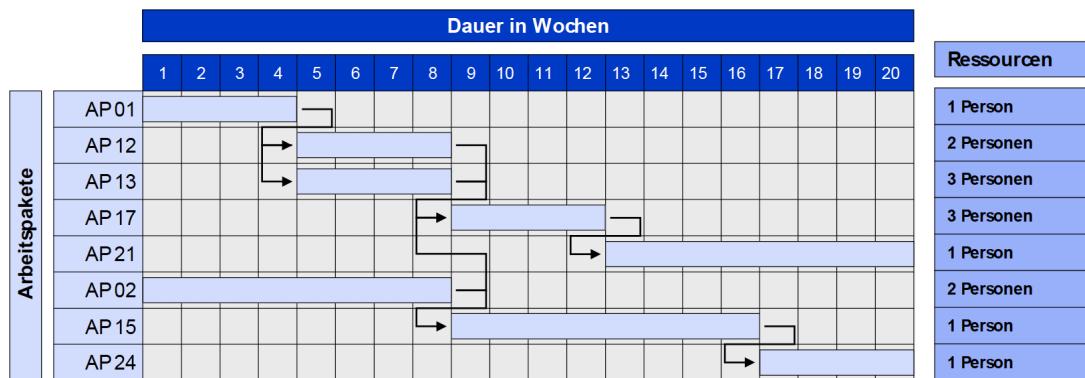


Abbildung 16: Beispiel Kapazitätsbelastung

Aus obiger Abbildung lässt sich nun ein Kapazitätsdiagramm erstellen, welches die notwendige Kapazität über die Projektdauer visualisiert. Somit wird ersichtlich, wann im Projekt wieviel Kapazität eines bestimmten Skills einer Personalressource verfügbar sein muss.

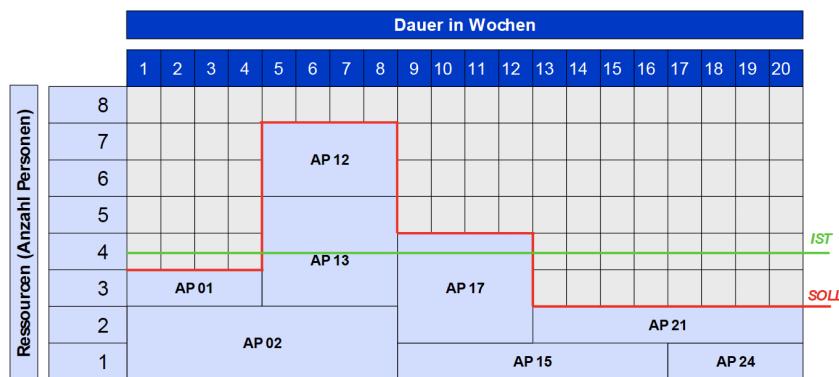


Abbildung 17: Beispiel Kapazitätsbelastungsdiagramm

Lustige Info: Im Skript *Projekt in Gang halten* unter zugehörigem Kapitel ist eine übungsaufgabe, bei welcher mithilfe eines VPN und zugehörigen Netzplänen für frühste und späteste Lage jeweils ein Kapazitätsbelastungsdiagramm für frühste/späteste Lage erstellt wird. Ich bin zu müde dass jetzt noch hier reinzuklatschen.

2.7.5 Organisationsplanung / Aufbauorganisation

Input Ressourcenplan, Projektorganisation

Ziel Selektions Projektmitarbeitender, Überführung Ressourcenplanung in Aufbauorganisation

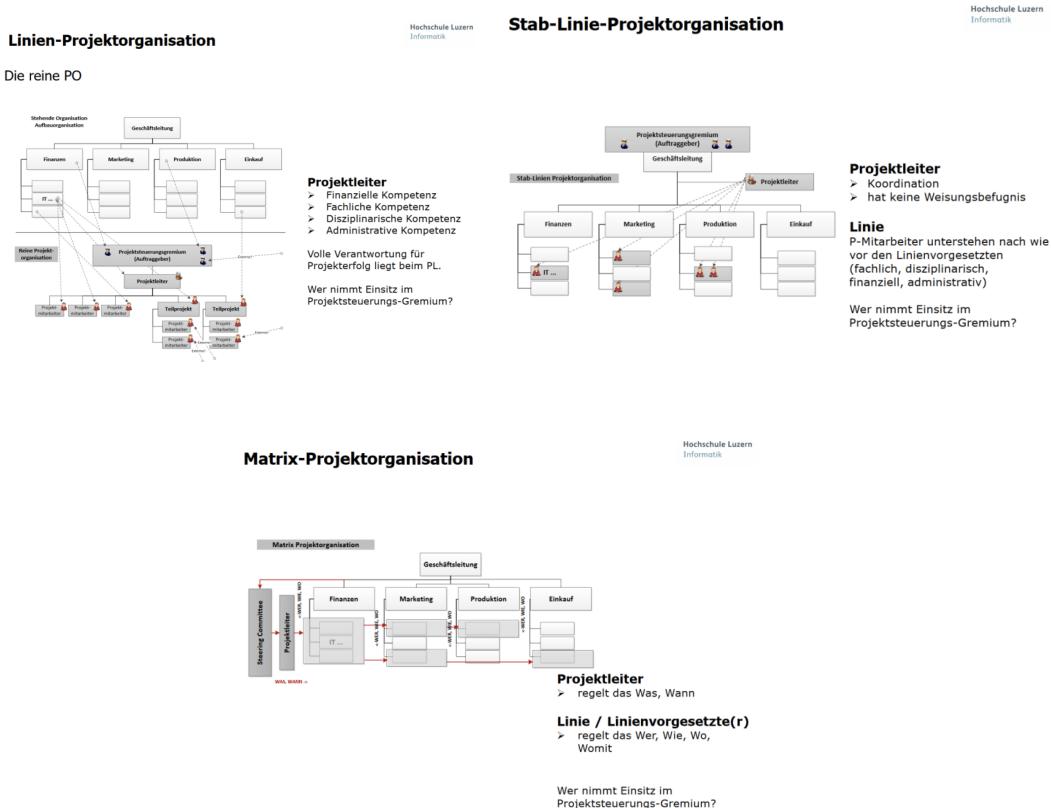
Output Mitarbeitende (intern, extern) zugeteilt auf APs

Lieferobjekt Organisationsplan

Tätigkeiten der Organisationsplanung:

- Projektmitarbeitende selektieren
- Projektorganisation initialisieren
- Stellebeschriebe → AKV : Aufgabe, Kompetenz, Verantwortung
- Arbeitspakete an Projektmitarbeitende zuordnen

Projektorganisationen 3 verschiedene Projektorganisationsarten:



Optimale Teamgrösse Optimale Teamgrösse basierend auf der Anzahl Kommunikationsbeziehungen in einer Gruppe.

- Pro Kommunikationsbeziehung: wöchentlicher Zeitverlust von 40 Minuten pro Mitarbeiter
- Arbeitszeit pro Woche & MA = 40 Stunden
- #MA = Anzahl Mitarbeiter

$$Dauer[\text{Tagen}] = \frac{\text{Aufwand}[\text{Tagen}]}{\#MA}$$

$$Dauer = \frac{1}{\#MA} \times \text{Aufwand}$$

$$Dauer = \frac{1}{\#MA \times \text{Produktivität}} \times \text{Aufwand} \quad (\text{Produktivität des MA berücksichtigt})$$

Team-Grösse (MA)	Kommunik.-beziehungen (KB)	Zeitverlust (ZV) durch Besprechung (40'/KB)	Produktive Restarbeitszeit / MA (PA) (40h – ZV)	Produktivität	Dauer= $1/(\#MA * PA) * \text{Aufwand}$ Aufwand = 1
3	3	2h	38h	38/40=95%	1/3*0.95=0.351
4	6	4h	36h	36/40=90%	0.278
6	15	10h	30h	75%	0.222
7	21	14h	26h	65%	0.220
9	36	24h	16h	40%	0.278
10	45	30h	10h	25%	0.400
11	55	37h	3h	7.5%	1.212
13					

Abbildung 19: Tabelle zur Berechnung der optimalen Teamgrösse

Aus der Tabelle ist ersichtlich: Bei einer Teamgrösse von 7 zeigt die Berechnung der Dauer einen Minimalwert von 0.22 → die kürzeste Dauer

2.7.6 Projektkostenplanung

Input Organisationsplan, Ressourcenplan

Ziel Ermittlung, Optimierung & Finanzierung der im Projekt anfallenden Kosten

Output Dokumentierte Kosten und deren Finanzierung

Lieferobjekt Projektkostenplan

Projektkosten werden die im Projekt anfallenden Kosten für das gesamte Projekt bzw. für die Gesamtheit aller AP verstanden.

Betriebskosten, welche nach Einführung anfallen, sind nicht Bestandteil der Projektkosten, müssen jedoch im Business Case aufgeführt sein.

- Fragen, die in **diesem Planungsschritt** beantwortet werden:
 - Was kostet das Projekt?
 - Welche Verluste fallen an, wenn das Projekt nach Konzeption/Realisierung/Einführung gestoppt wird (→ Ausblick auf Projektbudgetplan)?
 - Welche Kosten fallen intern/extern an?
 - Welche Kosten sind Investitionen, die amortisiert werden müssen?
 - Welche der möglichen Lösungsvarianten ist kostenmässig optimal (Kostenvergleichsrechnung)?
- Fragen, die in einem Business Case beantwortet werden müssen:
 - Was ist der Nutzen des Projekts? → Business Case
 - Was sind die Folgekosten (→ Nutzungsphase) des Projekts?
 - Welche Lösungsvariante ist über die gesamte Lebensdauer die Optimale (Kosten-Nutzen)?

Kostenarten & Kostenstruktur Mögliche Projektkosten:

- Interne Personalkosten
- Externe Personalkosten
- Material (HW, SW)
- Temporäre Mieten für Räumlichkeiten
- Interne Kosten für Bereistung & Betrieb von Testsystemen
- Zusätzliche Lizenzkosten pro-rata während Projektdauer
- etc.

Möglichkeiten für die Strukturierung der obigen Kostenarten:

- **Ausgabenwirksame Kosten vs. Interne Kosten:**

Ausgabenwirksame Kosten, die ohne Projekt nicht angefallen wären, bewirken einen Mittelabfluss. (bspw. Kosten für externe Projekt-MAs oder Investitionen)

Interne Kosten fallen nur kalkulatorisch an, sie fallen auch ohne Projekt an. (bspw. Kosten für interne Projekt-MAs, Umlagen anderer OEs auf Projekt o.ä.)

- **Kosten vs. Investitionen:**

Kosten, die während Projekt anfallen, beeinflussen Erfolgsrechnung, wobei **Investitionen** aktiviert werden (→ Anlagevermögen), sind somit bilanzwirksam und werden über bestimmte Zeitdauer abgeschrieben.

Kosten-Nutzen-Betrachtungen: Business Case (BC)

- Projektkostenplanung kann auch für Erarbeitung/Verfeinerung des BC genutzt werden
- Im BC wird direkter/indirekter, der (nicht) quantifizierbare Nutzen eines Projekts ausgewiesen
- BC betrachtet neben Projektphase auch explizit die Nutzungsphase (Gesamt-Lifecycle eines Systems)
- Positiver BC ist die unternehmerische Rechtfertigung für das Projekt!
- Direkter Nutzen:
 - Direkte Einsparungen
 - Vermeidung von Kosten
 - Erhöhung von Einnahmen
 - Nicht quantifizierbarer Nutzen
 - etc.
- Indirekter Nutzen:
 - Technische Vorteile
 - Marktauftritt
 - etc.

Variantenberechnungen

- Eine oder mehrere grobe Varianten einer Lösung vorhanden
- Erste Variantenberechnungen durchführbar, Vergleich dieser wird möglich
- Behandelte Variante: **Statische Kostenvergleichsrechnung**

Statische Kostenvergleichsrechnung

- Statische Methode, ermöglicht Kostenvergleich über gewisse Zeitperiode (üblich 1 Jahr) bspw. zwischen bisherigem & neuem System oder zwei neuen Systemen (Variante A & B)
- Nur Kosten berücksichtigen, keine Erlöse
→ beide Varianten haben denselben angenommenen Erlös / Nutzen

Kostenart	Var. A	Var. B
Investitionen / Kapitaleinsatz [CHF]	360'000.-	420'000.-
Betriebskosten, jährlich [CHF]	200'000.-	190'000.-
Nutzungsdauer = Abschreibungsdauer der Investition	4 Jahre	6 Jahre
Kalkulatorischer Zinsfuss	7%	7%

Abbildung 20: Kostenvergleich zwischen Variante A und B

Welche der beiden Varianten hat die tieferen jährlichen Gesamtkosten? (**Jahrl. GK**)

Jährliche Gesamtkosten **GK** setzen sich zusammen aus

JB jährliche Betriebskosten

JA jährliche Abschreibungen (auf getätigte Investitionen)

Z jährliche Zinskosten

$$Jahrl.GK = JB + JA + Z$$

JB sind Kosten, die für die Aufrechterhaltung des betriebsanfallen (bspw. Personalkosten, Strom, Räume, Wartung etc.)

→ im Beispiel: CHF 200'000 / CHF 190'000

JA Abschreibungen berechnen sich aus Investitionen/Kapitaleinsatz dividiert durch Abschreibungsdauer (fixe Dauer, manchmal gleichgesetzt mit Nutzungsdauer).

→ im Beispiel: $\frac{CHF\ 360'000}{4} = CHF\ 90'000$ / $\frac{CHF\ 420'000}{6} = CHF\ 70'000$

Z Jährlicher Zins wird auf Basis des eingesetzten Durchschnittskapitals ($\frac{Investition}{2}$) und dem kalkulatorischen Zinsfuss (durch CFO festgelegt) errechnet.

→ im Beispiel: $\frac{CHF\ 360'000}{2} \times 7\% = CHF\ 12'600$ / $\frac{CHF\ 420'000}{2} \times 7\% = CHF\ 14'700$

Kostenart	Var. A	Var. B
[JB] Jährliche Betriebskosten	200'000.-	190'000.-
[JA] Jährliche Abschreibung	90'000.-	70'000.-
[Z] Jährl. Zins auf durchschnittlichem Kapitaleinsatz Investition/2*7%	12'600.-	14'700.-
Jährliche Gesamtkosten [JB+JA+Z]	302'600.-	274'700.-

Abbildung 21: Jährliche Gesamtkosten der Variante A und B

Hinweis: Das Resultat gilt für Jahr 1 bis 4, ab Jahr 5 muss Var. A durch eine neue Lösung ersetzt sein.

2.7.7 Terminplanung

Input Ablaufplan, Projektkostenplan

Ziel Festlegung von Terminen (kalenderisch) für Meilensteine und APs (Start, Ende)

Output Konkrete Termin- und Einsatzplanung

Lieferobjekt Terminplan

Bisher flossen zeitliche Aspekte ein, es gab jedoch noch kein „Mapping“ auf kalenderischer Ebene. Bisheriger Plan wird auf Kalender gesetzt, Woche 1 erhält einen konkreten Starttermin. Es kommt zu **Planungsiterationen**: Zugeteilte Mitarbeiter sind abwesend, Auftraggeber nicht mehr zufrieden mit Wunschtermin etc.

Nach Terminplanung muss oft gesamte Planung (Aufwände, Ressourcen, Dauer) neu aufgesetzt werden, um obige Ansprüche zu berücksichtigen. Planungstools sind hier von Vorteil.

2.7.8 Projektbudgetplanung

Input Projektkostenplan, Terminplan

Ziel Abstimmung/Aufteilung Projektkosten auf effektiv verfügbare (finanzielle) Mittel und eventuelle Budgetrestriktionen

Output Kalendarische Finanzierung

Lieferobjekt Projektbudgetplan

Planung der Projektkosten mit effektiv (im Unternehmen) vorhandenen Mitteln abgeglichen.

Budgetrestriktionen, vor allem wenn ein Projekt über Rechnungsabschluss (Jahresende) dauert, können in diesem Planungsschritt eruiert & gelöst werden. Einbezug der Finanzabteilung ist entscheidend. Alle Projekt- und Folgekosten müssen in entsprechenden Budgets berücksichtigt sein. Vorallem Folgekosten fallen meist in den Kostenstellen der stehenden Organisation an, nicht in dedizierten Projektkostenstellen.

Budgetplanung stellt zudem sicher, dass bei Projekt, welches über das Jahresende hinweg läuft, korrekte Kosten in den richtigen Jahren budgetiert werden. Diese Kosten fallen ebenfalls in stehender Organisation an und müssen von Kostenstellenleitern in deren Budget berücksichtigt werden (Projektleiter hat Bring-Schuld!).

Budgetprozess für neues folgendes Jahr startet oft bereits Ende Sommer des laufenden Jahres (Vorlaufzeichen), was die Projektbudgetplanung weiter erschwert.

2.7.9 Informationsplanung

Input Organisationsplan, Terminplan

Ziel Sicherstellung der Versorgung aller Stakeholder mit relevanten Infos & Dokumentation

Output Kommunikations- & Informationskonzept

Lieferobjekt Informationsplan, Dokumentationsplan

Informationsplan Klarheit darüber, wann wer welche Informationen über welche Kommunikationskanäle in welcher Form zur Verfügung gestellt bekommt („**push**“). Sollte folgende Spalten enthalten:

- **Was** (Fortschrittsbericht monatlich, Requirements Katalog präsentieren etc.)
- **Verantwortliche Person** (Projektleiter etc.)
- **Wann** (Anfangs nächsten Monat, 5.3.2017 etc.)
- **Form** (Bericht, Präsentation etc.)
- **Empfänger** (Steuerungskomitee, Teilprojektleiter, Power User etc.)
- **Kanal** (Email, Meeting etc.)

Dokumentationsplan Zu klärende Fragen:

- **Dokubedürfnisse:**
Was und für wen muss dokumentiert werden?
- **Dokuzeitpunkt:**
Wann muss dokumentiert werden? Wie langt gilt eine Aufbewahrungspflicht?
- **Doku-Art / Doku-Träger:**
Wie muss dokumentiert werden? Wo wird die Doku aufbewahrt?
- Kategorien im Dokumentenplan:
 - Abwicklungsdocumentation (Projektauftrag, Phasenabnahmen etc.)
 - Benutzerdokumentation (Benutzerhandbuch, Krisenplan etc.)
 - Wartungs-/Betriebsdokumente (Systemarchitektur, Systembeschrieb etc.)
- Spalten für Dokumentenplan:
Dok-Nr., Dokumentationsart, Erstellzeitpunkt, Träger, Aufbewahrungsort

3 Project Management - Projekt in Gang halten

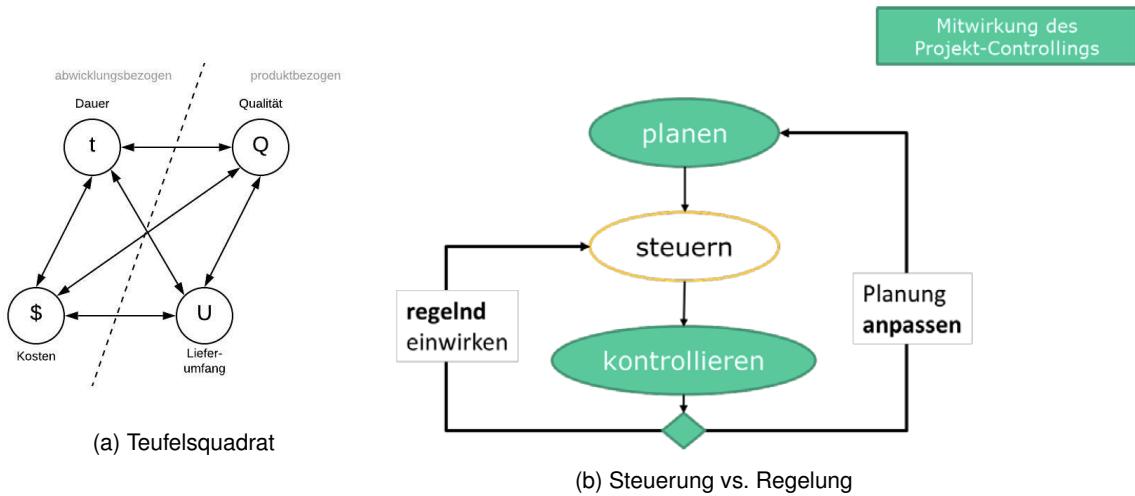


Abbildung 22: Teufelsquadrat & Steuerung vs. Regelung

- Planungswerte durch Dimensionen des Teufelsquadrat repräsentiert
- Planungseinheit, die Einfluss nehmen kann, ist ein definiertes AP
- Die 4 Dimensionen sind voneinander abhängig,
Änderung einer Dimension hat Reaktion auf anderer Achse zur Folge

Steuerung vs. Regelung In einem Projekt wird geregelt, nicht nur gesteuert. (Wenn wir ein Fahrzeug „steuern“, sind wir eig. regelnd unterwegs).

Zusammenhang zw. Projektplanung & Projektfortschritt überwachen Aus Planung entsteht Basisplan zu Zeitpunkt t_0 . Zu einem Zeitpunkt t_x wird der Fortschritt des Projekts dann kontrolliert/überwacht. Je nachdem wie kritisch die Abweichungen sind werden Massnahmen ergriffen und ein neuer Plan zum Zeitpunkt t_x erstellt (der wieder abgenommen werden muss).

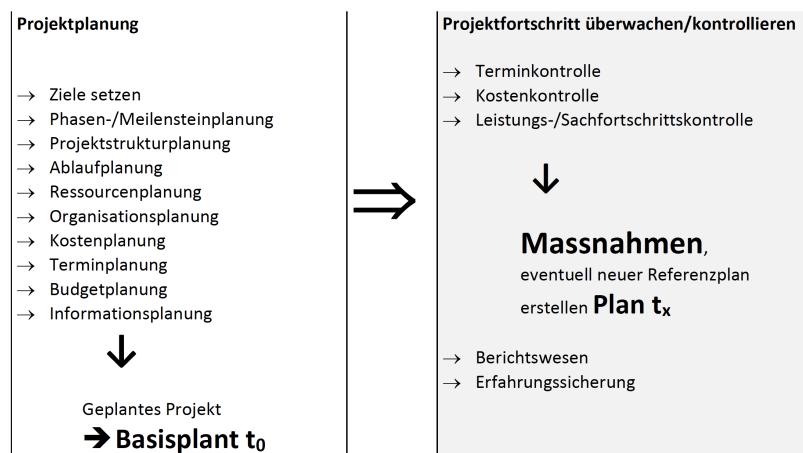


Abbildung 23: Zusammenhang Projektplanung / Projektfortschritt überwachen

3.1 Kontrollbereiche

- **Definition „messen“:**

Kann als kontrollieren/überwachen interpretiert werden; umfasst alle Aktivitäten, um projektbezogene Abweichungen zw. Plan- und Ist-Zustand aufzudecken

- Korrigierende Massnahmen sind Teil des Steuerns, nicht des Messens

- Unterscheiden:

- Planungskontrolle (Kontrolle bezieht sich auf Projektabwicklung)
- Realisierungskontrolle (Kontrolle auf das vom Projekt zu liefernde Produkt)

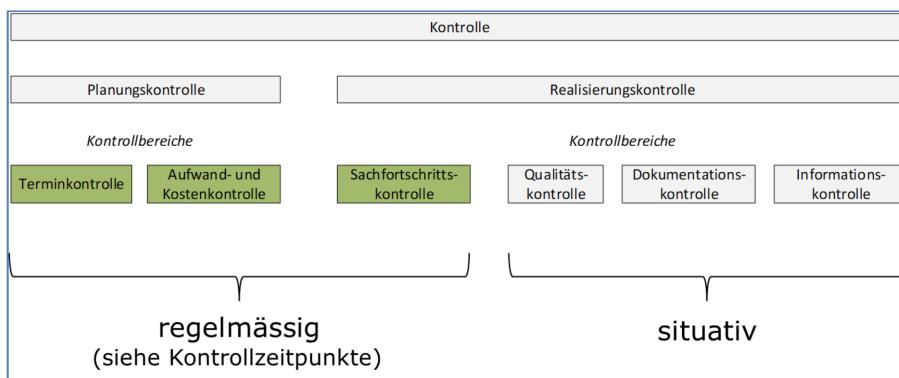


Abbildung 24: Kontrollbereiche in Projektabwicklung

- **Terminkontrolle**

Ziel Projektüberprüfung auf Einhaltung des geplanten Endtermins. Fokus auf kritische Pfade

Fragestellung Sind in Planung vorgegebene Termine (Meilensteine) eingehalten worden?

Prognose Wann wird das Projekt voraussichtlich beendet?

- **Kostenkontrolle**

Ziel Überprüfung des Projekts aus Einhaltung geplanter Kosten

Fragestellung Sind bis zum Kontrollzeitpunkt kumulierte Kosten eingehalten worden?

Prognose Zu welchen Kosten wird Projekt voraussichtlich abgeschlossen?

- **Leistungs-/Sachfortschrittskontrolle**

Ziel Produkt überprüfen auf Erreichung des geplanten Lieferumfangs

Fragestellung Wie weit ist Arbeit am Produkt fortgeschritten?

Prognose Wird Projektgegenstand wie abgemacht geliefert?

Kontrollzeitpunkte

- Festlegen der Kontrollzeitpunkte nach:

- **nach einer Zeitdauer oder Arbeitsvolumen**

nach vorgegebener Zeitdauer oder bei Erreichen eines Kontrollvolumens, egal ob bei Phasenende oder Resultat. Zeitdauer typischerweise zw. 1-3 Monaten, Volumen bei 100-200 geleisteten Personentagen.

- **bei Erreichung eines Resultates**

Lieferobjekt nach Erstellung möglichst unmittelbar kontrollieren, um allfällige Fehler schnell erkennen zu können

- **bei Erreichung von Phasenende/Meilenstein**

Phasenabschluss ist prädestiniert für einen Kontrollzeitpunkt

3.2 Terminkontrolle

- **Wieso Projekttermine kontrollieren?**

Endzeitpunkt des Projekts sicherstellen, erreichte Projekttermine müssen gesichert werden

- Termine haben mit Dauer eines Projekts zu tun, nur kontrollierbar wenn sie eindeutig benannt sind

- Typische Termine:

- Meilensteintermine
- Start- und Endtermine von APs
- Audit- & Reviewtermine
- Projektanfangs- und -endtermin
- Weitere wie Berichtstermine etc.

- Dauer aus Aufwand und verfügbaren Ressourcen:

$$Dauer[\text{Tage}] = \frac{\text{Aufwand}[PT]}{\text{Personen}[P]}$$

3.2.1 Termintreue (%)

Kennzahl, wie viele vergangene Termine eingehalten werden konnten oder eben nicht. Diese Kennzahl gibt Auskunft über Termintreue bzw. lässt aufschliessen, wie akkurat die Planung war (Aufwand & Dauer)

$$\text{Termintreue [%]} = \frac{\#\text{TermineEingehalten}}{\#\text{Termine}} \times 100[\%]$$

3.2.2 Zeitlicher Fortschrittsgrad (%)

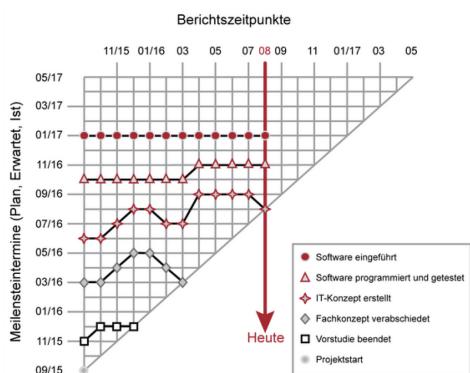
$$\text{Zeitl. Fortschrittsgrad}_{IST} [\%] = \frac{Dauer[Wochen]}{\text{prog. Gesamtdauer[Wochen]}} \times 100\%$$

$$\text{Freiheitsgrad PLAN: } \text{zeitFG}_{PLAN} = \frac{\text{Istdauer}}{\text{geplanteDauer}}$$

$$\text{Freiheitsgrad IST: } \text{zeitFG}_{IST} = \frac{\text{Istdauer}}{\text{voraussichtlicheDauer}}$$

(Abweichung zw. zeitFG_{PLAN} & zeitFG_{IST} : grobes Indiz für Einhaltung des Endtermins)

3.2.3 Termin-Trenddiagramm



Leselogik:

- Kurve nach oben, wenn Termine immer wieder nach hinten verschoben. Strebt weg von ursprünglich geplantem Termin (weg von der Diagonale)
- Kurvenverlauf horizontal, wenn keine Terminabweichungen erkennbar sind
- Kurve nach unten, wenn Termine vorverschoben werden können (strebt der Diagonale entgegen)

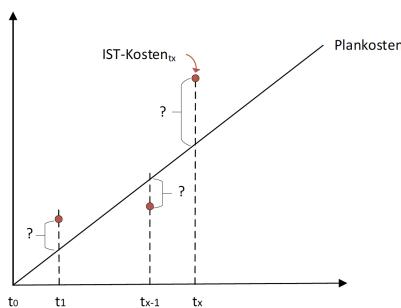
- **Vorteile:** einfach, schnell zu erstellen, Terminabweichungen sofort erkennbar, geeignetes Kommunikationsmittel (innerhalb/ausserhalb Projekt), Abstimmungsdefizite erkennbar, fördert Terminbewusstsein
- **Nachteile:** subjektive Einschätzung, Trendkurve allein nicht ausreichen zur Ursachenanalyse, Diagramm muss kommentiert sein

3.3 Kostenkontrolle (Aufwandkontrolle)

Wieso Projektkosten kontrollieren? Ziel der Kostenkontrolle ist, vereinbarten Kostenrahmen des Projekts sicherzustellen. Der gemäss Planung vorgesehene zeitliche Kostenanfall muss gesichert werden. Kosten können *time* oder *material* sein.

3.3.1 Ist- und Plankosten

- In der Kostenkontrolle werden zwei Kosten einander gegenübergestellt & verglichen:
 - IST-Kosten (zum Kontrollzeitpunkt t_x aufgelaufene effektive Kosten, kumuliert)
 - Plankosten (zum Zeitpunkt nach Planung verbrauchte Kosten, kumuliert)
- Es können 3 mögliche Zustände vorliegen (siehe Abbildung 25a):
 - IST-Kosten liegen zum Zeitpunkt t_x unter dem geplanten Verbrauch
 - IST-Kosten entsprechen zum Zeitpunkt t_x dem geplanten Verbrauch
 - IST-Kosten liegen zum Zeitpunkt t_x über geplantem Verbrauch
- Es interessieren primär Personalkosten (reflektieren effektiv geleistete Aufwände)



(a) schematische Kostenkurve

		Fortschritt ist ...		
		über Plan	gemäß Plan	unter Plan
IST-Kosten sind ...	über Plan	?	rot	rot
	gemäß Plan	grün	grün	rot
	unter Plan	grün	grün	?

(b) Bewertungsbereich von Kosten

- Zum Zeitpunkt t_x mehr Ressourcen verbraucht als geplant
Zum Zeitpunkt t_{x-1} jedoch weniger als geplant
- Zustand bei t_x so interpretierbar, dass Projekt weiter fortgeschritten als geplant, da offenbar mehr Aufwände rapportiert als geplant
- Oder dass trotz Mehrkosten resultativ weniger erreicht wurde als bei t_x geplant
- Kostenkontrolle muss deshalb unbedingt den **Sachfortschritt** des Projekts miteinbeziehen

3.3.2 SOLL-Kosten (Earned Value)

Aufgelaufene IST-Kosten zum Kontrollzeitpunkt t_x reflektieren geleisteten (Personen-)Aufwand im Projekt. Ist aber nicht gleichbedeutend mit Fortschritt bezogen auf das zu erstellende Endresultat (Produkt). Von Interesse ist, was mit rapportiertem Aufwand geleistet wurde! Kostenkontrolle muss immer auch den Leistungsstand in Betracht ziehen.

Arbeitswert, SOLL-Kosten: Entspricht Kosten, die in einem AP bei einem bestimmten Fortschrittsgrad (FG) nach Plan hätten anfallen dürfen. Earned Value kann maximal den Planwert (Planned Value PV) des APs erreichen.

Beispiel

- Arbeitspaket:
 - Plankosten** CHF 10'000
 - IST-Kosten** CHF 7'500
 - Fortschrittsgrad** 25%
- AP wird in der Hälfte seiner Dauer überprüft, zum Zeitpunkt t_x sind IST-Kosten CHF 7'500
- FG ist aber nur bei 25% (nicht wie erwartet bei 50%, bei Hälfte der Dauer)
- Es hätten nur $\frac{1}{4}$ der Plankosten (also CHF 2'500) anfallen dürfen
- SOLL-Kosten** liegen somit bei CHF 2'500

3.4 Leistungskontrolle / Sachfortschrittskontrolle

- **Wieso Leistungsfortschritt kontrollieren?**

Soll den Lieferumfang des Projekts sicherstellen.

- Dazu müssen zum Zeitpunkt t_x vorgesehene Leistungen, Resultate & Fortschritt gesichert werden

3.4.1 Methoden zur Bestimmung des Fortschrittsgrads

Zur Berechnung des Arbeitswertes (SOLL-Kosten) muss Fortschrittsgrad zum Kontrollzeitpunkt t_x bestimmt sein. Dazu bieten sich folgende Möglichkeiten an:

- **Meilensteinmethode**

Anzahl erreichte Meilensteine vs. Total aller Meilensteine

- **Relativer Fertigstellungsgrad / Befragung**

relative Betrachtung der Fertigstellung; Ermittlung einer genauen %-Grösse, sehr subjektiv

- **0/50/100%-Methode**

Betrachtung, welche noch nicht gestartetes AP mit 0%, begonnenes AP mit 50% und abgeschlossenes AP mit 100% bewertet

- **0/100%-Methode**

Betrachtung, welche erst bei Abschluss eines AP 100% vergibt, alle laufenden / nicht begonnenen APs werden mit 0% bewertet

- **Restaufwandschätzung**

zukunftsorientierte Methode; Statt erbrachte Leistung zu bewerten wird Restaufwand geschätzt (wieviel Aufwand muss noch geleistet werden), Subtraktion des Restaufwands von geplantem Aufwand ergibt eine Abschätzung des Fortschritts (unter Berücksichtigung eingesetzter Ressourcen)

Methode	Voraussetzung	Beurteilung
$\frac{\text{erreichte Meilensteine} \times 100}{\text{alle geplanten Meilensteine}}$	Meilensteine müssen genügend differenziert vorliegen.	Führt für viele Projekte zu verlässlichen Ergebnissen.
$\frac{\text{Istaufwand} \times 100}{\text{Planaufwand}}$	Aufwand muss ein Indikator für die Leistung sein.	Geringe Anforderungen an die Daten, auch bei undifferenzierter Planung anwendbar.
0/50/100-Methode, 0/100-Methode	Arbeitspakete dürfen nicht zu groß sein und sollten nicht zu lange dauern.	Führt für viele Projekte zu verlässlichen Ergebnissen.
$\frac{\text{Istaufwand} \times 100}{\text{Voraussichtlicher Gesamtaufwand}}$	Restaufwand muss zuverlässig geschätzt werden können.	Führt für viele Projekte zu verlässlichen Ergebnissen.
$\frac{\text{Istkosten} \times 100}{\text{Voraussichtliche Gesamtkosten}}$	Restkosten müssen zuverlässig geschätzt werden können.	Führt für viele Projekte zu verlässlichen Ergebnissen.

Abbildung 26: Möglichkeiten zur Bestimmung des Fortschrittsgrads

Beispiel Gegeben ist ein Arbeitsplan mit APs A bis F. Für jedes ist effektiver Leistungsfortschritt indiziert (100% oder 25%). Auch PLAN- und IST-Kosten sind für das Projekt bekannt:

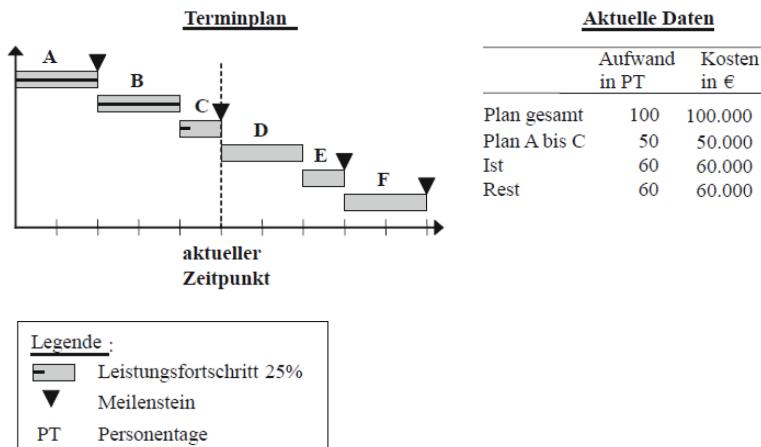


Abbildung 27: Terminplan mit aktuellen Daten

Die Arbeitswerte (erwirtschafteter Wert) für die APs lässt sich nach 0/50/100-Methode folglich berechnen:

Arbeitspaket	Plan-aufwand	Tatsächlicher Fortschritt	Fortschritt nach der 0/50/100-Methode	Erwirtschafteter Wert
A	20	100 %	100 %	20
B	20	100 %	100 %	20
C	10	25 %	50 %	5
D	20	0 %	0 %	0
E	10	0 %	0 %	0
F	20	0 %	0 %	0

Abbildung 28: Berechnung des Fortschritts & erwirtschaftetem Wert

3.5 Integrierte Kontrollsicht - Leistung-/Kosten-Kontrolle

Kosten reflektieren geleisteten Personenaufwand, ist jedoch nicht gleichbedeutend mit Fortschritt bezogen auf das zu erstellende Endresultat (Produkt). Kostenkontrolle muss auch Leistungsstand (Fortschritt) mit einbeziehen.

→ Earned-Value-Methode

3.5.1 Earned Value (EV)

- Neben PLAN- und IST-Kosten auch SOLL-Kosten (Earned Value) betrachten
- Synonym: *Budgeted Cost of Work Performed - BCWP*
- EV** entspricht denjenigen **Kosten, die für Erreichung des eruierten FG hätten anfallen dürfen**
- EV eines AP zum Kontrollzeitpunkt t_x berechnen:

$$EV_{tx} = FG_{tx} \times \text{Plankosten}_{AP}$$

(EV eines AP kann nie höher sein als die Plankosten eines AP)

3.5.2 Earned Value Methode (EVM)

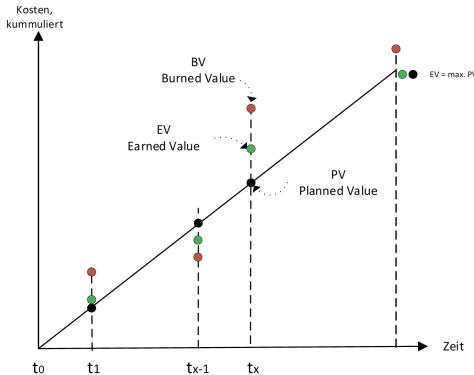


Abbildung 29: PV, BV und EV auf der Zeitachse

Kennzahlen:

- **Burned Value (BV)**

Kumulierte IST-Kosten (auch *Actual Cost (AC)* oder *Actual Cost of Work Performed (ACWP)*)
Herkunft: aus der Rapportierung

- **Planned Value (PV)**

Kumulierte PLAN-Kosten (auch *Budgeted Cost of Work Scheduled (BCWS)*)
Herkunft: aus der Planung

- **Earned Value (EV)**

Wie erläutert (auch *Budgeted Cost of Work Performed (BCWP)*)

Herkunft: berechnet aus eruiertem Fortschrittsgrad (%) bei t_x mal Plankosten des AP

- **Scheduled Performance Index (SPI)**

Zeitliche Kenngrösse, sagt aus ob Projekt vor oder hinter dem Plan liegt

Berechnung: $SPI = \frac{EV}{PV}$

SPI = 1 Projekt ist planmäßig unterwegs

SPI > 1 Erbrachte Leistung grösser als geplant, Projekt schneller als geplant

SPI < 1 Erbrachte Leistung kleiner als geplant, Projekt langamer als geplant

– Terminprognose mit SPI: $progDauer = \frac{Dauer.geplant}{SPI}$

- **Cost Performance Index (CPI)**

Kostenbezogene Kenngrösse, sagt aus ob Projekt über oder unter Plan läuft

Berechnung: $CPI = \frac{EV}{BV}$

CPI = 1 Projekt kostenmäßig gemäss Plan unterwegs

CPI > 1 Erbrachte Leistung grösser als tatsächliche Kosten, Projekt kostengünstiger als geplant

CPI < 1 Erbrachte Leistung kleiner als tatsächliche Kosten, Projekt teurer als geplant

– Kostenprognose mit CPI: $progKosten = \frac{Kosten.geplant}{CPI}$

3.5.3 Beispiel einer Anwendung der EVM

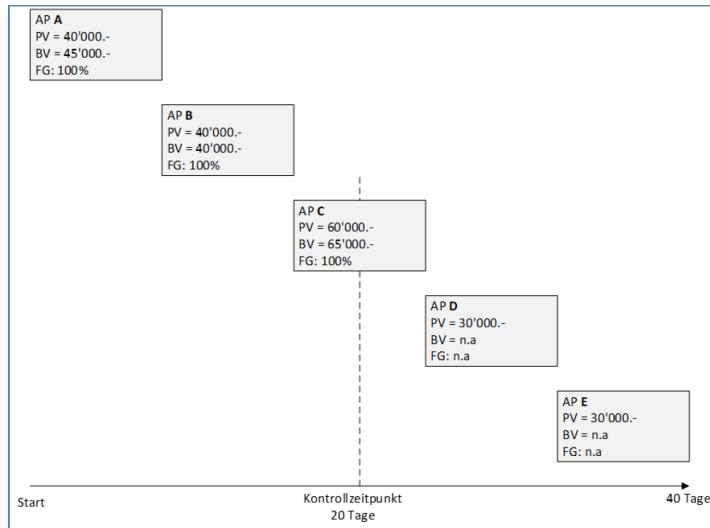


Abbildung 30: Übersicht von AP mit jeweiligen Werten

- **PV** bei t_x : $40'000 + 40'000 + \frac{60'000}{2} = 110'000$ CHF
- **BV** bei t_x : $45'000 + 40'000 + 65'000 = 150'000$ CHF
- **EV** bei t_x : $100\% \cdot 40'000 + 100\% \cdot 40'000 + 100\% \cdot 60'000 = 140'000$ CHF
(EV berechnet sich auf Plankosten des gesamten AP)
- **SPI**: $\frac{EV}{PV} = \frac{140'000}{110'000} = 1.3$
prog. Endtermin: $\frac{\text{Dauer.geplant}}{SPI} = \frac{40}{1.3} = 31$ Tage
- **CPI**: $\frac{EV}{BV} = \frac{140'000}{150'000} = 0.9$
prog. Kosten: $\frac{\text{Kosten.geplant}}{CPI} = \frac{200'000}{0.9} = 215'000$ CHF

3.5.4 Erfahrungssicherung

- Aufgetretene Probleme werden nach *vermeidbar*, *kaum vermeidbar* und *nicht vermeidbar* aufgeteilt
- Fragen, die zu beantworten sind:
 - Erfahrungen in Zusammenarbeit mit Lieferanten, Kunden, Fachabteilungen etc.?
 - Qualität der Teamarbeit?
 - Zweckmässigkeit des gewählten Vorgehen im Projekt?
 - Bewährten sich die eingesetzten Instrumente?
 - Angemessene Arbeitspakete (Aufwand, Dauer)?
 - Kompetenzen & Verantwortung richtig zugewiesen?

3.5.5 Reporting / Berichtswesen

Berichtswesen und Projektdokumentation obliegen der Fortschrittskontrolle. Ein akkurate Berichtswesen muss folgende 3 Kriterien erfüllen:

- Aktualität
- Empfängerbezogenheit
- Entscheidungsorientierung

3.6 Projekt steuern

3.6.1 „Adding People to a late project makes it even later“

Was, wenn Projekt in Schieflage gerät? Massnahmen gefragt, welche sind jedoch wirksam?

Wieso? Möglich ist, dass Kommunikations- bzw. Abstimmungsaufwand entlang Teamgrösse wächst. Je mehr Mitglieder einem Team zugehören, umso tiefer ist individuelle Produktivität der Einzelnen.

→ Kapitel zu *Optimale Teamgrösse!*

3.6.2 Dynamik der Projektarbeit

Projekt unterliegt dynamischen Einflüssen und Feedback Loops.

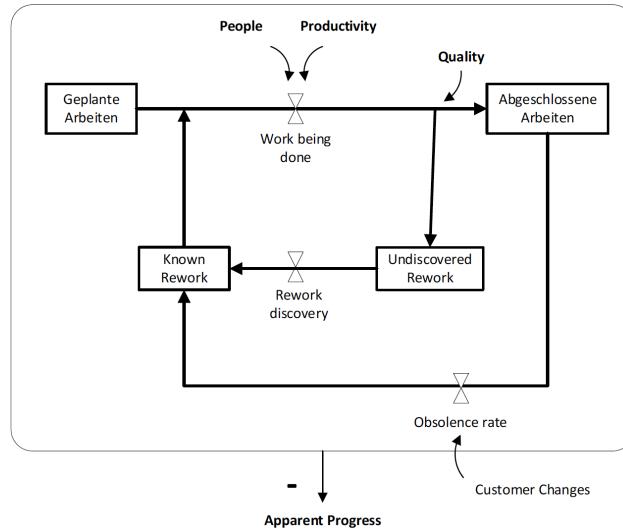


Abbildung 31: Projektablauf mit dynamischen Einflüssen

Korrekt geplante Projekte (ohne Reserven) können nach dieser Logik immer ins Hintertreffen kommen. Customer Changes (obsolescence rate) reduzieren Fortschritt (apparent progress) massgebend. Change Management ist ein Muss!

Korrigieren durch Massnahmen wie bspw. **Hiring** (weitere Personen ins Projekt), **Overtime** (Überzeit anordnen), **Schedule Acceleration** (Arbeitstakt beschleunigen) etc.

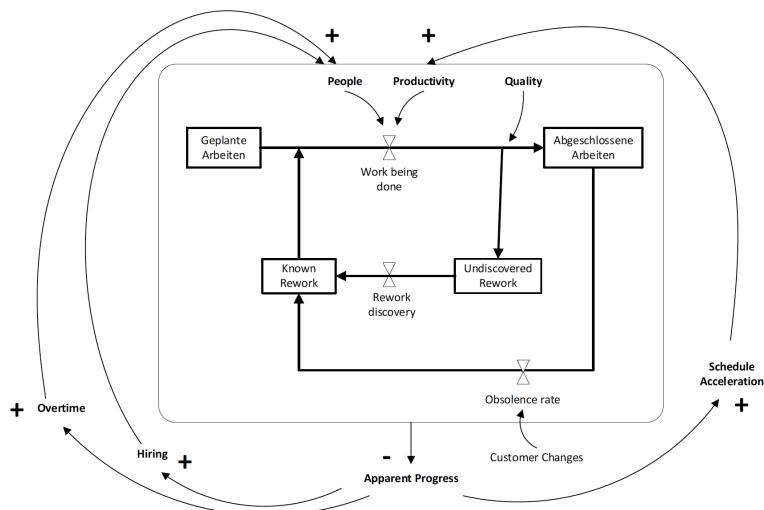


Abbildung 32: Projektablauf mit korrigierenden Massnahmen

Ausgleichende Massnahmen haben jedoch Wechselwirkungen (lösen Reaktionen aus)

- **Overtime:** mehr Überzeit, desto grösser wird Müdigkeit jedes Projektmitarbeitenden, Burn-Out-Rate steigt, was direkten Einfluss auf Produktivität/Qualität hat, sie sinkt
- **Hiring:** nominelle Stärkung der Belegschaft, aber durchschnittlicher Ausbildungsstand tiefer, führt zu Qualitätseinbussen
- **Schedule Acceleration:** Einfluss auf Qualität & Produktivität, da Koordinationsaufwand grösser wird, Engpässe entstehen, Motivation sinkt

Es entstehen *self-reinforcing Loops*, welche verstärkend wirken

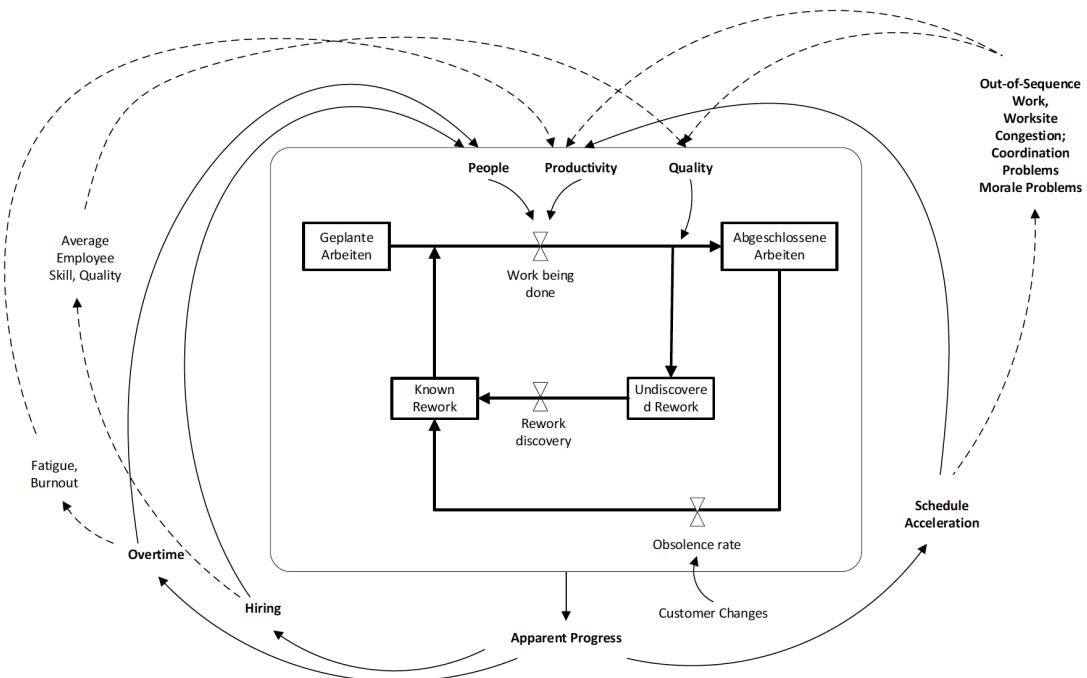


Abbildung 33: Einfluss der Wechselwirkungen auf Projektablauf

Projektleiter muss getroffene Massnahmen sehr kritisch hinterfragen und effektive Wirksamkeit beurteilen.
Gilt besonders für Personalmassnahmen („*Adding People to a late project makes it even later*“)

4 Requirement Engineering - Einführung / Setting

4.1 Ziel

- Ziel: angestrebter Zustand / erwünschte Wirkung, zukünftige Ergebnisse welche durch bestimmte Massnahmen / Lösungen erreicht werden sollen
- Ziele sollen vor Ermittlung der Anforderungen bekannt sein
- Ziele beschreiben übergeordnete Erwartungen an Erfolg eines Projekts
- Jede einzelne Anforderung muss sich auf ein Ziel zurückführen lassen
- Eine Zieldokumentation enthält (aus Pohl/Rupp S. 67 ff.)
 - Ziel-Nummer
 - Zielname
 - Zielwert
 - Messgröße
 - Zeitraum

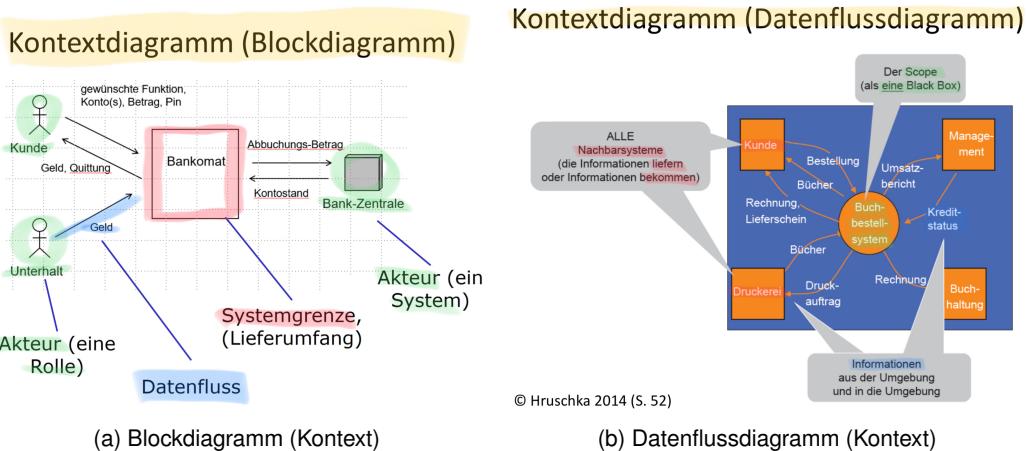


Abbildung 34: Kontextdiagramme

Typische Fehler in einem Kontextdiagramm

- Zusammenhänge ausserhalb, die das System gar nicht betreffen
- Unspezifizierte Ein- und Ausgaben im System
- Nicht identifizierte Nachbarsysteme

4.2 Stakeholder

- Werden in einer Stakeholdertabelle festgehalten
- Benötigt werden dabei pro Stakeholder mindestens:
 - Seine Rolle im System
 - Name
 - Wissen
 - Interessen

Es gilt ebenfalls zu klären: Was ist die **Aufgabe** des Stakeholders, wofür ist der Stakeholder **verantwortlich**, welche **Weisungsbefugnis** hat der Stakeholder, welche **Ziele** sollen vom Stakeholder erreicht werden, welche **Freigaben** kann der Stakeholder zu geben, auf welche Informationen hat der Stakeholder **Zugriff**, über welche **Kommunikationswege** werden Informationen ausgetauscht, wann erhält der Stakeholder **Feedback** zu seinen gelieferten Informationen, welche verantwortlichen **Personen** werden in welchen **Situationen** vom Stakeholder informiert, mit wem **koordiniert** der Stakeholder bestimmte Anforderungen, welche **Zugriffsrechte** erhält der Stakeholder für das Anforderungsrepository, für welche Fachgebiete ist der Stakeholder offizieller **Ansprechpartner**?

4.3 RE - Komponenten

Requirement Engineering setzt sich aus R-Development (Entwicklung) und R-Management (Verwaltung) zusammen. Anforderungen können dabei sein: Geschäftprozesse, Geschäftsfälle, Standards, technische Rahmenbedingungen, Funktionalitäten, qualitative Bestimmungen, Geschäftsregeln, Prototypen, Screens etc.

→ was zur Festlegung der zu entwickelnden Dienstleistung gebraucht wird

- Development setzt sich zusammen aus

Elicitation (Erheben, Ermitteln) welche Erhebungstechniken eignen sich für welche speziellen Informationsbedürfnisse? Vorgehen, wenn man etwas in Erfahrung bringen will/soll?

Analysis (Analyse, Auswertung) wie Resultate der Erhebung auswerten? (Strukturieren, Zusammenhänge herstellen, Lücken aufdecken etc.)

Specification (Spezifikation, Dokumentation) Welche Artefakte, Dokumente, Modelle, Ergebnisse etc. sollen angelegt werden, um das System adäquat zu beschreiben? Verständlich und sinnvoll für die Stakeholder?

Validation (Validierung) Wie eignet man sich mit entscheidungsrelevanten Stakeholdern auf eine zu realisierende Version? (Prüfen & Priorisieren der Anforderungen etc.)

- **Management (Verwaltung):** Wie / wo werden alle notwendigen Dokumente der Anforderungsentwicklung verwaltet, um einen Überblick zu behalten?

5 Requirement Engineering - Anforderungen ermitteln

5.1 Klassifizieren von Stakeholder-Inputs

Bedürfnisse (Anforderungsinformationen) müssen verschiedenen Kategorien zugeordnet werden:

- **Geschäftsanforderungen**

Alles, was finanziellen/geschäftlichen Vorteil auf dem Markt beschreibt, das Kunden oder Unternehmen mit dem Produkt erzielen möchten. Kann bspw. mit einem Wert assoziiert werden (*Der Marktanteil wird um X% erhöht*)

- **Anwendungsfälle und -szenarien**

Allgemeine Aussagen zu Benutzerzielen/Geschäftsausgaben, die die Benutzer ausführen müssen. Kann mittels eines Anwendungsszenarios dargestellt/ausgedrückt werden, bspw. *Ich möchte ein Adressetikett für ein versendetes Paket ausdrucken können*

- **Geschäftsregeln**

Wenn nur bestimmte Benutzerklassen unter bestimmten Bedingungen einer Aktivität nachgehen können, bspw. *Chemiker soll nur Chemikale von Gefahrenstufe 1 bestellen können, wenn er für diese Stufe geschult wurde*

- **Funktionelle Anforderungen**

Beschreiben erkennbares Verhalten des Systems unter bestimmten Bedingungen sowie Tätigkeiten, die ein Benutzer mithilfe des Systems verrichten kann, bspw. *Steigt der Druck über 40 bar, muss das entsprechende Warnlich aufleuchten*

- **Qualitätsmerkmale**

Geben Aufschluss darüber, wie gut das System bestimmte Aufgaben ausführt.

- **Anforderungen an die externe Schnittstelle**

Beschreiben die Schnittstelle zwischen einem System und dem Rest der Welt.

- **Einschränkungen**

Einschränkungen in Design & Implementierung begrenzen verfügbare Optionen des Entwicklers.

- **Datendefinitionen**

Beschreiben Vorgabewerte, Datentyp oder Formate, die vom Stakeholder gewünscht/verlangt werden.

- **Lösungsideen**

Konkrete Verfahren oder Anforderungen, welche eine Interaktion mit dem System beschreiben, welches erforderlich ist, um bestimmte Tätigkeiten zu verrichten.

5.2 Ermitteln

- Was wissen wir? (bekanntes Wissen)
- Was wissen wir noch nicht? (unbekanntes Wissen)
- Wie bringen wir „Nichtwissen“ in Erfahrung? Welche Anforderungsquellen/Erhebungstechniken?
- Wie dokumentieren wir unser Wissen, was soll/muss dokumentiert werden?
- Wie überprüfen/validieren wir unser Wissen?

5.3 7 Product Dimensions

						
User	Interface	Action	Data	Control	Environment	Quality Attribute
Users interact with the product	The product connects to users, systems, and devices	The product provides capabilities for users	The product includes a repository of data and useful information	The product enforces constraints	The product conforms to physical properties and technology platforms	The product has certain properties that qualify its operation and development

Abbildung 35: Die 7 Produktdimensionen

User Wer initiiert eine Aktion und was sind die Ziele? Wer erhält Output des Produkts?

Interface Welche Schnittstellen senden Daten/Nachrichten zu oder aus dem Produkt?

Action Welche Aktivitäten erfüllen die Bedürfnisse des Nutzers?

Data Welche Daten benötigen die Nutzer vom Produkt?

Control Welches sind die Policies, Regulationen und Geschäftsregeln, die das Produkt erzwingen muss?

Environment Wo wird das Produkt genutzt? Welche Software/Hardware wird genutzt, um das Produkt zu entwickeln/bedienen?

Quality Attribute Welches sind die Service-Stufen für Availability, Installability, Usability etc.?

5.4 W-Fragen für bspw. Interviews

Ziel(e) ↓ Aufgabe(n) ↓ Verrichtung Objekt	Aufgabenträger	Sachmittel	Zeit	Ort	Menge
Was, wie ? Woran ?	Wer ?	Womit ?	Wann, wie lange ?	Wo, woher, wohin ?	Wie oft, wieviel ?
Aufgabenmerkmale		Merkmale der Aufgabenerfüllung			

Abbildung 36: Einteilung der W-Fragen

K.R.O.K.U.S. Regel zur Formulierung von Fragen

K Kurze Fragen stellen

R Redundante Fragen vermeiden

O Offene Fragen stellen

K Konkrete Fragen stellen

U Unterfragen (& Kettenfragen) vermeiden

S Suggestive Fragen vermeiden

6 Requirement Engineering - Anforderungen dokumentieren

Zu Beginn muss man sich über die zu beschreibenden Eigenschaften einigen. Der Verwendungszweck bestimmt, ob ein Modell geeignet ist oder nicht.

6.1 Drei komplementäre Perspektiven

Dabei gibt es drei komplementäre Perspektiven zu beachten:

Struktur Relevante Objekte & ihre Beziehungen untereinander modellieren

→ Informations- / Datenmodell (Business Object Model)

Funktion Abläufe der Manipulation von relevanten Informationen modellieren

→ Aktivitätsdiagramm / Datenflussdiagramm

Verhalten Erlaubte Systemzustände & Zuständsänderungen aufgrund

eintretender Ereignisse modellieren

→ Zustandsdiagramm

Beispiel Perspektiven an Containerschiff

Struktur Schiff, Ladung, Besatzung, Hafen etc.

Funktion Ladung löschen, Anlegen, Fortbewegen, Laden, Schiff in Hafen lotsen etc.

Verhalten Zustandsübergänge → angelegt, auf hoher See, geladen, entladen etc.

6.2 Sichten der Anforderungsformulierung

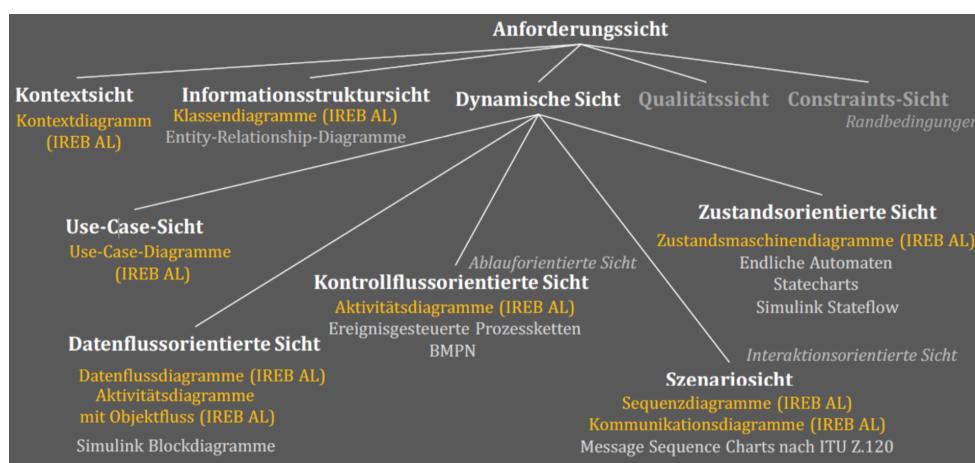


Abbildung 37: Anforderungssichten

6.3 Diagramme im RE

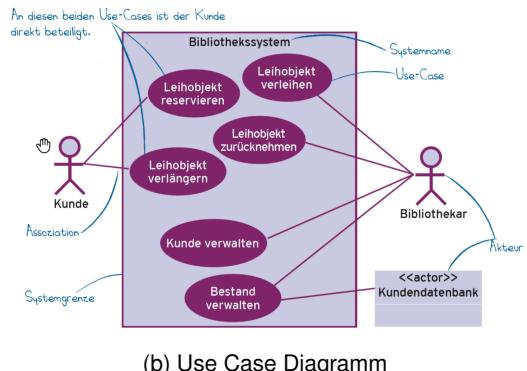
Perspektive	Diagramm
Struktur	Klassendiagramm
Systemfunktionen	Use Case Diagramm
Funktionsdetails	Aktivitätsdiagramm (UC-Spezifikationsschablone)
Verhalten	Zustandsdiagramm Sequenzdiagramm
System / Kontext	Kontextdiagramm

Abbildung 38: Diagramm-Möglichkeiten im RE

6.3.1 Beispiele von Diagrammen

Name	Leihobjekt verleihen
Kurzbeschreibung	Der Use-Case beschreibt, welche Prüfungen in welcher Reihenfolge nötig sind, damit ein Kunde ein Leihobjekt aus der Bibliothek ausleihen kann.
Akteure	Bibliothekar
Vorbedingungen	Der Bibliothekar ist am System angemeldet
Fachlicher Auslöser	Der Bibliothekar initiiert die Funktion Leihobjekt verleihen
Hauptszenario	Kunde identifizieren Entleiheberechtigung prüfen Leihobjekt identifizieren Entleihebarkeit Leihobjekt prüfen Verleihung Leihobjekt an Kunden prüfen Ausleihen speichern Beleg ausdrucken

(a) Use Case Beschreibung



(b) Use Case Diagramm

Abbildung 39: Komponenten eines Use Case

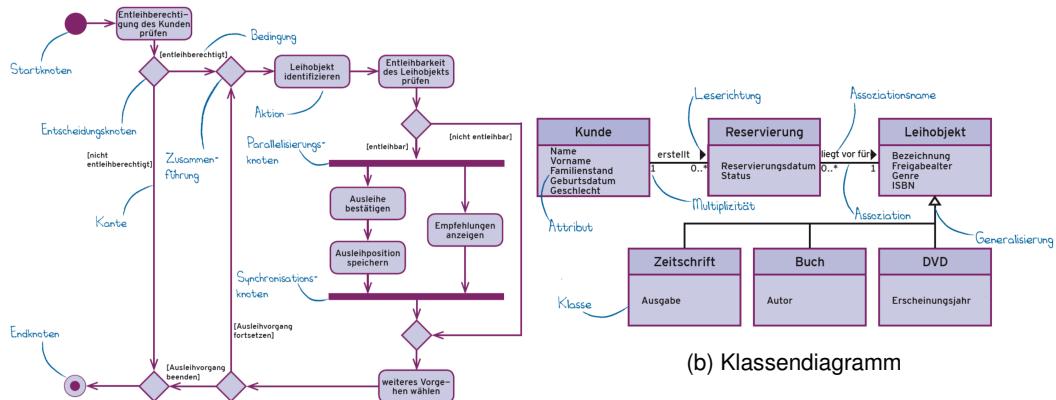


Abbildung 40: Aktivitäts- und Klassendiagramm

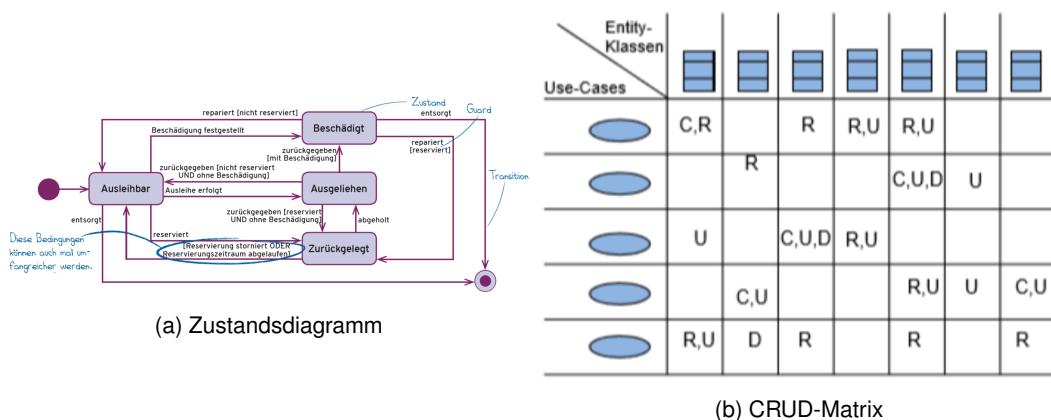
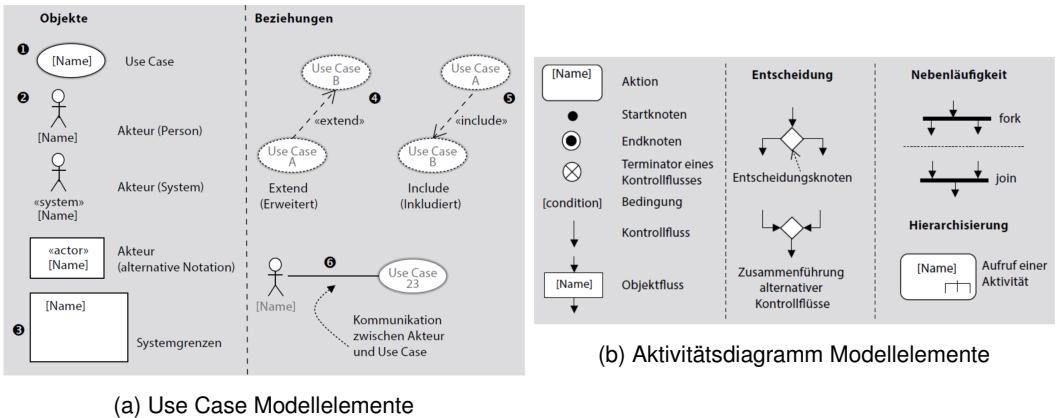


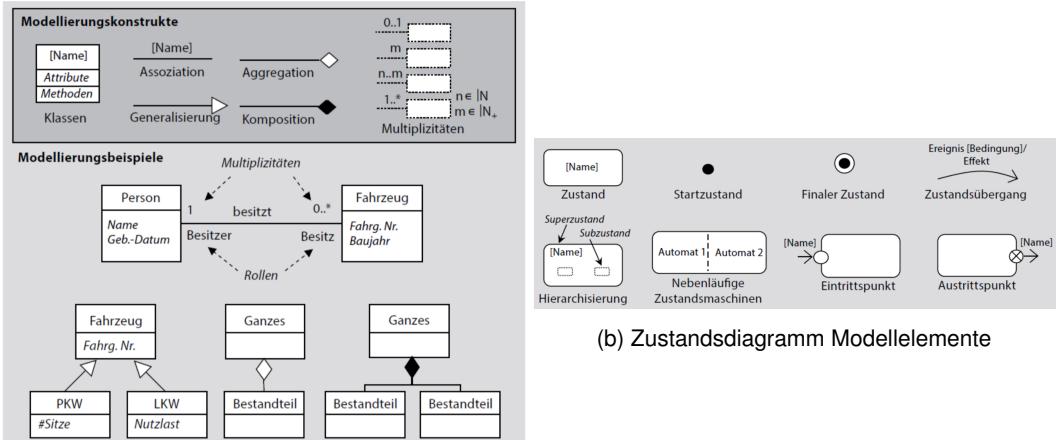
Abbildung 41: Zustandsdiagramm und CRUD-Matrix

6.3.2 Modellelemente verschiedener Diagramme



(a) Use Case Modellelemente

Abbildung 42: Modellelemente Use Case & Aktivitätsdiagramm



(a) Klassendiagramm Modellelemente

Abbildung 43: Modellelemente Klassen- und Zustandsdiagramm

Use Case

- Informationen sammeln - woher soll man das wissen?
- Mögliche Akteure finden - Wer arbeitet mit dem System?
- Mögliche Anwendungsfälle - Welche Ziele werden mit dem System erreicht?
- Akteure & Anwendungsfälle verbinden - Wer tut was mit dem System?
- Akteure beschreiben - Wofür steht ein Akteur?
- Weitere Anwendungsfälle suchen - Wie unterstützt das System die Akteure?
- Anwendungsfälle bearbeiten - Was gehört wirklich zu Anwendungsfällen?
- Anwendungsfälle dokumentieren - Was passiert im Anwendungsfall?
- Beziehungen zw. Anwendungsfällen modellieren - Was ist wieder verwendbar?
- Sicht validieren - Alles richtig?
 - Ist Anwendungsfalldiagramm vollständig?
 - Sind alle Anwendungsfälle korrekt?
 - Ist der Detailierungsgrad zweckmäßig?
 - Sind Anwendungsfälle richtig benannt?
 - Sind Akteure korrekt?
- Mögliches **Abstrahieren** von Akteuren

CRUD-Matrix Ist quasi die Gegenüberstellung von Use Cases (Funktionen) und Klassen/Entities (Datensstrukturen). Validierung der Struktur- und Funktionsperspektive. Datenzugriff wird über die Funktionalität eines Use Cases bestimmt.

Aktivitäts-, Klassen-, Zustandsdiagramm Das haben wir jetzt schon so oft, die ganze Zeit, gehabt. Das sollte man mittlerweile ungefähr im Griff haben.

6.4 Anforderungen in Textform

Regeln

- Anforderungen stets in vollständigen Sätzen schreiben
- Immer im Aktiv formulieren
- Begriffe konsistent verwenden, Synonyme & Hynonyme vermeiden
- Glossar erstellen, darin definierte Begriffe verwenden
- Prozesse durch Vollverben formulieren

Transformationsprozesse

- Nominalisierung
- Substantive ohne Bezugsindex
- Universalquantoren
- Unvollständig spezifizierte Bedingungen
- Unvollständig spezifizierte Prozesswörter

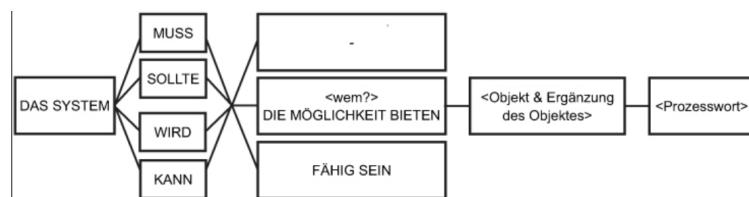


Abbildung 44: Vollständige Satzschablone

Entscheidungstabelle Anwendung bei unvollständig spezifizierten Bedingungen.

R(X) mögliche Fälle

Bedingungen können erfüllt oder nicht erfüllt sein

Aktionen Definieren, bei welchen Bedingungen was (nicht) ausgeführt werden soll

Tabellenbezeichnung	R1	R2	R3	R4	R5	R6	R7	R8
Bedingungen								
Lieferfähig	j	j	j	j	n	n	n	n
Angaben vollständig	j	j	n	n	j	j	n	n
Bonität in Ordnung	j	n	j	n	j	n	j	n
Aktionen								
Lieferung mit Rechnung	x		x					
Lieferung als Nachnahme		x		x				
Angaben vervollständigen			x	x				
Mitteilen: nicht lieferbar					x	x	x	x

Abbildung 45: Beispiel einer Entscheidungstabelle

7 Requirement Engineering - Agiles RE

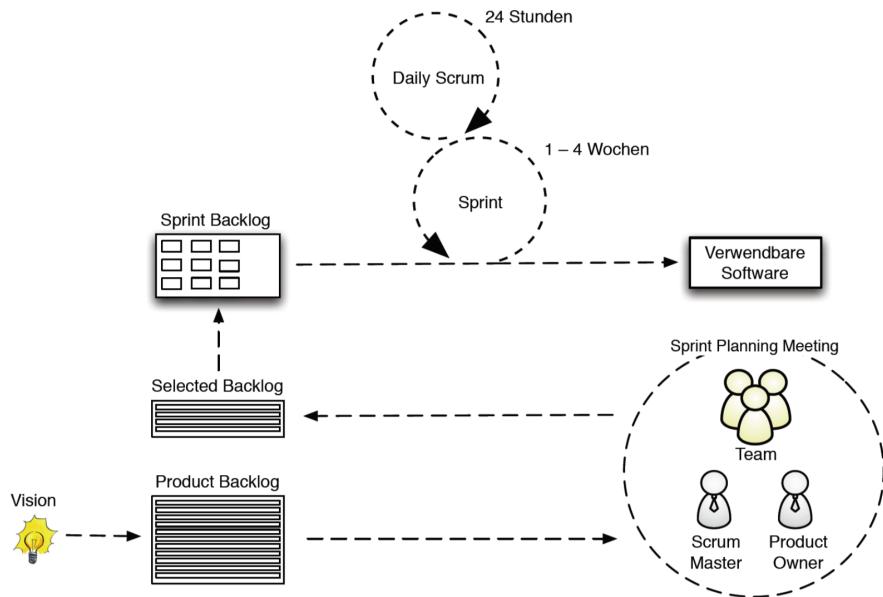


Abbildung 46: SCRUM im Überblick

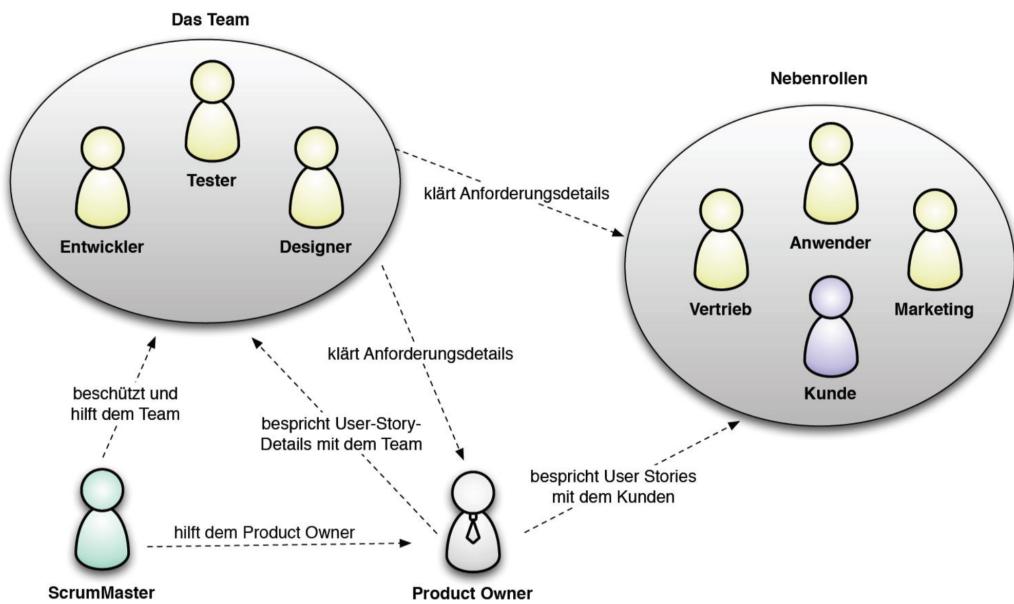


Abbildung 47: Zusammenspiel zwischen den SCRUM-Rollen

Product Backlog Beinhaltet User Stories, welche umgesetzt werden sollen

User Story Task User Story und ihre dazugehörigen Anforderungen/Ziele

Sprint Backlog Zu bearbeitende User Stories im Sprint, offene/in Arbeit/fertig

7.1 Dokumentation von Anforderungen in agilen Projekten

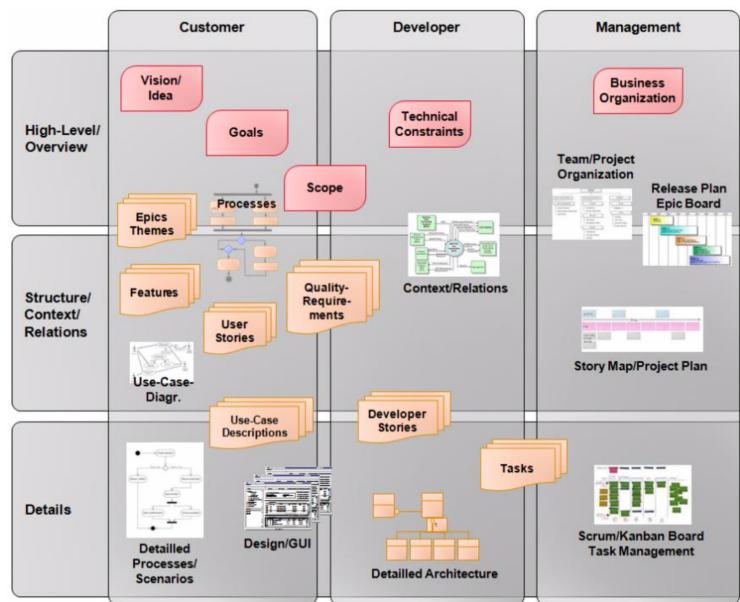


Abbildung 48: Überblick über alle Requirements-Artefakte

7.1.1 Backlog Items in agilem Projekt

- Backlog Items dienen
 - zur Strukturierung & Einordnung von Anforderungen
 - zur Release-Planung
 - als Speicher für detaillierte Anforderungen
 - zur Aufwandschätzung (beim Zuteilen zu einem Sprint)
- *Backlog Refinement*: zur Umsetzung von Backlog Items ist oft eine Verfeinerung / Detaillierung der User Stories notwendig

7.1.2 User Story Mapping

- Landkarten für das Produkt Backlog
- Bietet die Möglichkeit, gesamtes Produkt/Dienstleistung als Reihe von Aufgaben zu betrachten, die der Benutzer erfüllt
- Praktisch gesehen: Raster von User Stories aufbauen, mit Überschriften anordnen (soll die Erfahrung des Benutzers beim Durchlaufen des Produkts darstellen)



Abbildung 49: Vorgehen beim Erstellen einer Story Map

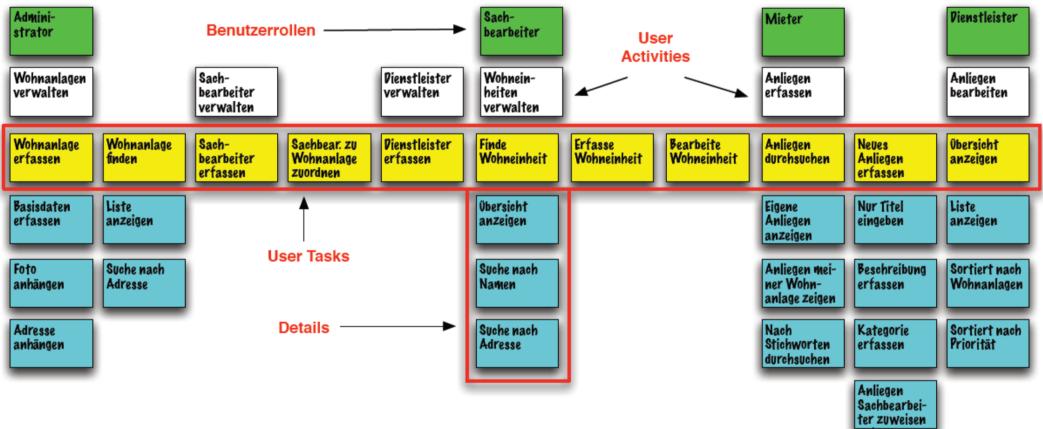


Abbildung 50: Beispiel einer Story Map

7.1.3 Scrum User Story

- Kurzer, narrativer Text, der Interaktion zw. Benutzer & System beschreibt
- Gilt als Grundlage für Kommunikation als Ausgangspunkt für Umsetzung von Produktfunktionalitäten
- User Stories werden im Scrum Product Backlog verwaltet
- Dienen als Planungsinstrument für die Sprints
- Bestandteile einer User Story:
 - **Name:** kurze prägnante Benennung
 - **Beschreibung:**
 - * Wer? - *Als [Rolle/Akteur]*
 - * Was? - *möchte ich [Funktionsbeschreibung]*
 - * Warum? - *damit [Begründung]*
 - **Akzeptanzkriterien:** Wie wird festgestellt, ob User Story entsprechend der Anforderungen umgesetzt wurde? Beantwortet die Frage, was getestet werden soll
Die Funktion ist korrekt implementiert, wenn [messbare Kriterien]
 - **Ergänzende Beschreibungen:** typisch Link auf Use Case oder andere Anforderungsformen

I.N.V.E.S.T.-Kriterien für User Stories

I Independent

Story ist unabhängig von anderen Stories. Darf nicht davon abhängen, dass andere Stories zuerst umgesetzt werden müssen.

N Negotiable

Kunden & Entwickler erarbeiten und präzisieren User Stories gemeinsam. Kunde beschreibt grob Funktionalität, Details werden mit Entwickler ausgearbeitet.

V Valuable

Stories sollen erkennbaren Mehrwert für betroffene Nutzergruppen liefern. Möglicher Weg ist, Stories gemeinsam mit jeweiligen Benutzern zu verfassen.

E Estimatable

Story muss überschaubar sein, dass Entwickler die Umsetzung der Anforderung schätzen können. Entwickler müssen auch über fachliche & technische Kompetenzen verfügen.

S Small

Konkreter Umfang von User Stories muss von Team entschieden werden. Stories können zu gross/klein sein. Regel: Story soll min. halben Personentag, max. zehn Personentage erfordern.

T Testable

Tests bilden Massstab, ob eine Story erfolgreich abgeschlossen wurde oder nicht. Testbarkeit muss zwingend gewährleistet sein.

7.2 Scrum Backlog Items

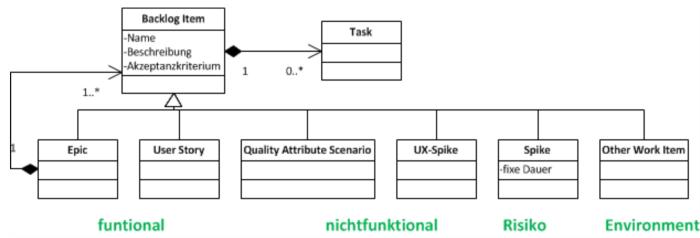


Abbildung 51: Übersicht über Scrum Backlog Items und deren Zugehörigkeiten

7.2.1 Tasks

- Teilaufgaben innerhalb von Backlog Items:
Werden durch Team vor der Umsetzung definiert, dienen der Arbeitsteilung
- Dienen häufig der personenbezogenen Zuordnung (Zuordnung meist auch fachlich bedingt)
- Tasks verbessern die Aufwandschätzung
- Backlog-Items (besonders User Stories) besitzen oft einheitliche Tasklisten
(bspw. nach Disziplinen aufgeteilt: *Architektur erweitern, GUI, Datenbank anpassen, etc.*)

7.2.2 Epics

- Grobgranulare Anforderung, die zu komplex/umfangreich sind, um geschätzt oder in einem Sprint umgesetzt zu werden
- Meist nicht nach User Story Schema formuliert
- Müssen in andere Backlog-Items zerlegt werden (bspw. User Stories / Quality Attribute Scenario)
→ User Story Splitting (Aktivität des RE in Scrum → Backlog Refinement)
- Unterscheidung versch. Arten von Epics
 - Business Epics**
 - fachliche Anforderungen aus Geschäftswelt
 - beschreiben organisations-/prozessübergreifende Eigenschaften
 - Beispiel: *Die Zeiterfassung wird effiziente Erfassungs- und Auswertemöglichkeiten für alle Zeitarten (Tagesarbeitszeit, Fehlzeiten etc.) bereitstellen.*
 - Architectural Epics**
 - weitreichende technische Vorgaben
 - betreffen hauptsächlich Architekturfragen
 - Beispiel: *Die neue SW-Lösung für die Arbeitszeiterfassung wird auf allen Typen von Endgeräten zur Verfügung stehen und wir werden damit technologisch führend sein.*

7.2.3 Quality Attribute Scenario

- Mess- und testbar formulierte, nichtfunktionale Anforderungen (**Qualitätsanforderung**)
 - Source** (Benutzer, Computer, System etc.)
 - Stimulus** zu betrachtende Bedingung
 - Environment** In welchem Zustand befindet sich das System
 - Artifact** Welcher Teil des Systems wird beeinflusst
 - Response** Systemantwort
 - Response Measure** Messbares Attribut der Antwort (bspw. Zeit)
- Beispiele:
 - Response Time:
When customers click check-out button, order info appears in ≤ 2 seconds
A validity check may not take more than 5 seconds.
 - Scalability:
Access to product catalog must scale up to 10'000 concurrent users that experience the maximum response times stated above

7.2.4 Spikes / UX-Spikes / Other Work Items

Spike Time-boxed AP zur Minimierung von Risiken (bspw. Abklären Realisierungsmöglichkeiten)

UX-Spike APs zur Erarbeitung & Überprüfung der User Experience (bspw. Prototyping)

Other Work Item Arbeititem zum Aufbau & Pflege der Arbeitsumgebung

7.3 User Story Splitting

- In einem Epic sind meist mehrere zusammenhängende User Stories (funktionale Anforderungen) und Quality Attribute Scenarios (nichtfunktionale Anforderungen)
- Nach Aufteilung/Verfeinerung von Epics ist realistische Sprint Planung möglich
- Jede neu erzeugte User Story muss weiterhin einen Kundennutzen darstellen
- 9 Patterns für eine praktische Herangehensweise des User Story Splittings:
 - **Workflow Schritte** (Workflow Steps)
evtl. Minimalworkflow heraustrennen, später erweitern, nur Beginn und Ende des Workflow etc.
 - **Performance nachlagern** (Defer Performance)
Erst funktional, danach nichtfunktionalen Anteil nachliefern (bspw. Performance)
 - **Simpel / Komplex** (Simple/Complex)
Erst einfacher Kern, später die Erweiterungen
 - **Grösster Aufwand** (Major Effort)
Hauptaufwand herauspicken, später weitere Ausprägungen
 - **Variation der Schnittstellen** (Data Entry Methods)
Erst Daten einer Schnittstelle, danach die der anderen verarbeiten
 - **Variation der Daten** (Variations in Data)
Erst einen Teil der Daten, danach die restlichen Teile verarbeiten
 - **Variation der Geschäftsregeln** (Business Rule Variations)
Erst eine Teilmenge der Regeln, später die erweiterten Regeln umsetzen
 - **Operationen** (Operations [e.g. CRUD])
Mehrere Operationen (verwalten, konfigurieren etc.) in separate Stories aufteilen
 - **Einen Spike herausbrechen** (Break out a Spike)
Explorative Story schreiben, die mit minimalem Aufwand zur Klärung von Fragen führt, wiederholen

8 Requirement Engineering - Qualitative Anforderungen

8.1 Anforderungsarten

Ziele beschreiben Zustände

Funktionen beschreiben Fähigkeiten (Verben)

Qualitäten beschreiben Eigenschaften (Attribute)

Randbedingungen schränken den Lösungsraum ein

8.2 Vorgehen beim Klassifizieren von Anforderungen

Wird die Anforderung gestellt, weil...

- Systemverhalten, Daten, Eingaben oder Reaktionen auf Eingaben zu spezifizieren sind - unabhängig davon, wie dies geschehen soll?
→ **funktionale Anforderung**
- Restriktionen bzgl. Verarbeitungs-/Reaktionszeiten, Datenmengen oder Datenraten zu spezifizieren sind?
→ **Leistungsanforderung**
- eine spezielle Qualität, die das System aufweisen soll, zu spezifizieren ist?
→ **besondere Qualität**
- irgend eine andere Restriktion zu spezifizieren ist?
→ **Randbedingung**

8.3 Anforderungstypen ISO/IEC 25030

		Funktionale Anforderungen	
Software-Produktanforderungen	Inhärente Produkteigenschaften	Software-Qualitätsanforderungen	Interne Qualitätsanforderungen
		Software-Qualitätsanforderungen	Externe Qualitätsanforderungen
			„Quality in Use“ Anforderungen
Realisierungsanforderungen	Externe Produkteigenschaften	Produktmanagementanforderungen (z.B. Kosten, Anbieter, Lieferdatum)	
		Prozessanforderungen	
		Organisatorische Anforderungen	

Abbildung 52: Software- und Realisierungsbedingungen nach ISO/IEC 25030

8.4 Quality Model

Qualitätsmerkmale sind in zwei Modelle organisiert:

- **Quality In Use Model:** Qualität eines Systems aus Sicht des Anwenders
- **Product In Use Model:** Qualität des Systems als inherante Eigenschaft

Qualitätsmerkmale dienen als Grundlage für Ableitung von Qualitätsanforderungen.

8.4.1 Quality In Use Attributes

- Effectiveness** (*Effektivität*) Exaktheit und Vollständigkeit, mit welcher Anwender bei Verwendung des Systems ihre spezifischen Ziele erreichen
- Efficiency** (*Effizienz*) Betrachtet aufgewendete Ressourcen zur Erreichung der Ziele. Nach Standards umfasst dies auch Zeit (wie lange bis Anwender Ziel erreicht), Material (welche Materialien muss Anwender bereitstellen) oder Kosten (welche Kosten entstehen für Anwender bei Nutzung)
- Satisfaction** (*Zufriedenheit*) Grad der Zufriedenheit des Anwenders bei Systemnutzung. Anwender soll Erfahrung machen, dass System nützlich ist, er ihm vertrauen kann und dass Arbeit mit dem System ein angenehmes Erlebnis ist.
- Freedom of Risk** (*Risikofreiheit*) Grad der Verhinderung von Risiken bei Nutzung des Systems. Welche Massnahmen das System ergreift, um Risikofolgen gering zu halten.
- Context coverage** (*Kontextabdeckung*) Grad der Abdeckung für ursprünglich gedachten Anwendungsbereich, Flexibilität um System ausserhalb des Anwendungsbereich zu nutzen.

8.4.2 Product in Use Attributes

- Functional Suitability** (*Funktionelle Eignung*) Setzt erwartete Funktionalität in angemessener/korrektter Art und Weise unter Berücksichtigung festgelegter Randbedingungen/Eigenschaften um. Fragestellungen nach Angemessenheit von Funktionalitäten für vorgegebenen Einsatzbereich.
- Performance Efficiency** (*Performanz & Effizienz*) Leistungsniveau in Abhängigkeit eingesetzter Betriebsmittel und geforderter Bedingungen. Zeitverhalten der Anwendung, Verbrauchsverhalten in Bezug auf Speicher, Prozessorzeit & weitere Ressourcen
- Compatibility** (*Kompatibilität*) Explizite Zusammenarbeit mit anderen Systemen und extreme Auswirkungen auf Systeme gleicher HW
- Usability** (*Benutzerfreundlichkeit*) Qualitätsmerkmal, misst wie gut Benutzerfreundlichkeit, um Nutzer bei effizienter/effektiver Zielerreichung zu helfen. Aufwand für Erlernung/Bedienung des Systems. Attraktivität des Systems für Benutzer.
- Reliability** (*Zuverlässigkeit*) Leistungsniveau unter festgelegten Bedingungen über festgelegten Zeitraum zu bewahren. (bspw. Robustheit gegenüber fehlerhaften Eingaben, Stabilität bei Umgang mit fehlerhaften Systemzuständen und Wiederherstellbarkeit bei Systemausfällen)
- Security** (*Sicherheit*) steigende Bedeutung; alle in Bezug auf Datensicherheit revelante Teilmerkmale (Datenintegrität, Authentizität, Vertraulichkeit)
- Maintainability** (*Wartbarkeit/Änderbarkeit*) interne Qualität des Systems; Aufwand zur Analyse eines Fehlers/Fehlverhaltens, Aufwand zur Durchführung von Modifikationen mit Testbarkeit, Stabilität des Systems gegenüber unerwarteten Seiteneffekten.
- Portability** (*Portabilität*) Aufwand, System strukturell zu verändern, um auf andere Umgebung zu versetzen oder einzelne Systemkomponenten auszutauschen. Teilmerkmale sind Anpassbarkeit des Systems an untersch. Umgebungen, Aufwand zu Installation und Austauschbarkeit von Komponenten.

8.5 Abnahmekriterien

- Metrik** Explizites Verfahren zur Messung der Eigenschaft mit Kennzahlen
- Akzeptanzkriterium** Grenzwerte oder Grenzbereiche, die bei einer Messung die Erfüllung einer Eigenschaft nachweisen
- Operationalisierung** Optionale Angabe; Testszenarien/Testumgebung, kann Erfüllung der Eigenschaft sinnvoll mit vertretbarem Aufwand prüfen

8.5.1 Beispiele von Metriken

Funktionelle Eignung Vorgaben für Genauigkeit von Berechnungen

Performanz/Effizienz Antwortzeit, Durchsatz, Speicherverbrauch

Kompatibilität Anzahl Verletzungen von Standardvorgaben in Schnittstellen

Benutzbarkeit Durchschnittszeitaufwand Schulungen, Anzahl Clicks zur Zielerreichung

Zuverlässigkeit Verfügbarkeit (Gesamt(ausfall-)zeit)

Sicherheit Anzahl erfolgreich abgewendete vordefinierte Misuse Cases

Wartbarkeit Vollständigkeitsgrad Doku, Transitive Abhängigkeiten/Ähnlichkeit von Methoden

Portabilität Anzahl nicht-konfigurierbare Systemeigenschaften im Code

8.6 Quality Grid

Raster zur Bewertung von Qualitätsmerkmalen in einem System. Dabei werden die „Product in Use“ Merkmale nach ihrer Wichtigkeit bewertet. Dabei kann der Grad der Wichtigkeit bewertet werden mit:

- kritisch
- wichtig
- gemäss Firmenstandard (as usual)
- unwichtig
- vernachlässigbar

(Vor allem bei unwichtig/vernachlässigbar sind Begründungen hilfreich oder unerlässlich!)

9 Requirement Engineering - Anforderungen prüfen

Warum Anforderungen überprüfen?

- **Risiko:**
 - Juristisch: Entwicklungsauftrag auf Basis eines Anforderungsdokument, wenn dies falsch oder unvollständig ist, kann es zu Streitigkeiten führen
 - Fehlerfortpflanzung: Fehler in Anforderungen pflanzen sich in der Umsetzung fort. Späte Entdeckung → hohe Kosten
- **Ziel:**
 - Fehler (mehrdeutige/widersprüchliche/unvollständige Anforderungen) möglichst früh im Entwicklungsprozess entdecken
- **Anforderungsfreigabe:**
 - Weitere Entwicklung mittels festgelegter Prüf- und Abnahmekriterien entscheiden

9.1 Überprüfen von Anforderungen

Hauptziele beim Überprüfen von Anforderungen:

- **Inhalt**
Alle relevanten Anforderungen im erforderlichen Detaillierungsgrad erfasst?
- **Dokumentation**
Anforderungen gemässt festgelegten Dokumentations-/Spezifikationsvorschriften dokumentiert?
- **Abgestimmtheit**
Alle Stakeholder zufrieden mit dokumentierten Anforderungen, alle bekannten Konflikte aufgelöst?

Prüfkriterien „Inhalt“

- Vollständigkeit
 - alle relevanten Anforderungen erfasst?
 - alle für eine Anforderung notwendigen Informationen erfasst?
- Verfolgbarkeit
 - alle relevanten Anforderungsquellen erfasst?
- Korrektheit / Adäquatheit
 - spiegeln Anforderungen die Bedürfnisse der Stakeholder wider?
- Konsistenz
 - stehen Anforderungen nicht miteinander in Widerspruch?
- Keine vorzeitigen Entwurfsentscheidungen
 - keine Lösungen in den Anforderungen vorweggenommen?
- Überprüfbarkeit
 - Abnahme- bzw. Prüfkriterien definiert?
- Notwendigkeit
 - trägt jede Anforderung zur Erfüllung eines definierten Ziels bei?

Prüfkriterien „Dokumentation“

- Konformität zum Dokumentationsformat / Dokumentenstruktur
 - vorgegebene Schablonen / Modellierungssprache verwendet?
 - Struktur des Anforderungsdokuments eingehalten?
- Verständlichkeit
 - wurden verwendete Begriffe in einem Glossar definiert?
- Eindeutigkeit
 - textuelle Anforderung sprachlich eindeutig?
- Konformität mit Dokumentationsregeln
 - Syntax der Modellierungssprache eingehalten?

Prüfkriterien „Abgestimmtheit“ Im Verlauf der RE-Prozesses erwerben Stakeholder neues Wissen über das geplante System. Die Meinung über bereits abgestimmte Anforderungen kann sich ändern. Stakeholder können Änderungswünsche äußern, ohne nachfolgende Entwicklungsaktivitäten zu beeinträchtigen.

- Abstimmung
 - jede Anforderung mit allen relevanten Stakeholdern abgestimmt?
- Abstimmung nach Änderungen
 - für jede geänderte Anforderung erneute Zustimmung der Stakeholder eingeholt?
- Konflikte aufgelöst
 - alle bekannten Konflikte bzgl. Anforderungen aufgelöst?

Prinzipien der Prüfung von Anforderungen Diese verbessern die Qualität der Prüfungsergebnisse:

Prinzip 1 Beteiligung der richtigen Stakeholder

Prinzip 2 Trennung von Fehlersuche und Fehlerkorrektur

Prinzip 3 Prüfung aus unterschiedlichen Sichten

Prinzip 4 Geeigneter Wechsel der Dokumentationsform

Prinzip 5 Konstruktion von Entwicklungsaufgaben

Prinzip 6 Wiederholte Prüfung

9.2 Reviews - Begriff, Objekte, Typen & Arten

9.2.1 Reviews / Audits / User Acceptance Tests

Reviews are public, verbal, interpersonal examinations of an artifact (documentation, code etc.)

→ Product Quality

Audit is an impersonal verification of tangible evidence of a process such as documentation

→ Process Quality

Acceptance Tests typically involve blackbox testing, performed on a system

prior to transfer of ownership

→ Validation (*in Scrum misleadingly called review („Sprint Review“)*)

9.2.2 Reviews nach Objekten

- **Management Reviews** → Prozess

bspw. Review des Projektmanagement-Plans

- wird vorgehenskonform geplant?

- entspricht Projektverlauf der Planung?

- **Technical Reviews** → Produkt

bspw. Review der Systemspezifikation

- entspricht Architektur den Anforderungen?

- ist Architektur vernünftig umsetzbar?

- **Sprint Review** in Scrum, am Ende eines Sprints

→ Produkt-Inkrement wird untersucht

- war wurde im Sprint fertiggestellt?

- muss Product Backlog angepasst werden?

- *in SoDa spricht man von Sprint-Abnahme, da es eig. kein Review ist*

9.2.3 Reviews im Detail

Reviews und Meilensteine

- Reviews oft an Meilensteinen (oder als Vorbereitung dafür) durchführen

- Ein Meilenstein ist ein geplanter Punkt im Projektlauf, an dem vorher festgelegte (Zwischen-)Ergebnisse vorliegen, die es erlauben den Projektfortschritt zu messen

- Zu jedem Meilenstein gehören Artefakte: Dokumente, SW-Komponenten etc.

- Meilenstein ist erreicht, wenn geforderte Artefakte vorliegen und Review/Test erfolgreich war

Review Typen

Inspection strenger, wohldefinierter Prozess

Team Review geplant, strukturiert, etwas weniger formal

Walkthrough Autor stellt sein Werk einer Gruppe von Fachkollegen vor

Pair Programming zwei Entwickler arbeiten gemeinsam am selben Produkt

Peer Deskcheck nur eine Person prüft das Arbeitsergebnis

Passaround Kopien an versch. Personen versenden und deren Feedback einholen

Ad Hoc Review gewöhnlich normaler Bestandteil der Teamarbeit

Reviews im Projekt Lebenszyklus

Feasibility Review Systemdefinition & Projektplan

Requirements Review Requirementspez., User Management, System & Acceptance Test Design

Preliminary Design Review System Design & Intergrated Test Spezifikation

Critical Design Review Detailed Design & Unit Test Spezifikationen

Source Code Review

Product Release Review

Review - Dauer und Zeitpunkt

- Review-Sitzungen: maximal 1-2 Stunden
- pro Sitzung nur ein Objekt / Artefakt überprüfen (oder mehrere kleine verwandte)
- Klassisches Projektvorgehen: Reviews eng an Projektlebenszyklus gekoppelt, müssen dann angepasst werden, wenn zu überprüfende Objekte/Artefakte vorliegen sollten
- Besser Anforderungen & Konzepte reviewen statt Detail-Design & Code

Review - Psychologie Hauptsächlich zu beachten: Der erste Zug zu einem Review muss so aussehen, als käme er vom Autor und nicht vom Gutachter. Der Autor muss sich mit der Idee des Review wohlfühlen. Reviews sind vor allem eine zwischenmenschliche Angelegenheit, Erfolg hängt von der persönlichen Dynamik zwischen Autor und Reviewer ab.

- Flexibel sein, wer wessen Arbeit reviewed; kompatible Paare sich suchen lassen
- guter Reviewer ist nicht wertend & erkennt, dass der primäre Nutzen einer Review das gemeinsame Sprechen über ein Werk ist
- Mögliche / tatsächliche Fehler, die in Review gefunden werden, soll man möglichst lokal lösen (nicht an die grosse Glocke hänger und ein Trari Trara draus machen)

Review - Vorgehen & Stil

- Teilnehmer erhalten Objekte / Artefakte einige Tage vor der Review-Sitzung
- Alle Mitglieder des Review-Teams bereiten sich darauf vor (Unterlagen ernsthaft studieren)
- Leider der Review geht die Objekte / Artefakte durch und fragt Teilnehmer, ob sie Fragen bzgl. bestimmten Abschnitten / Seiten haben
- Entdeckte Fehler / aufgeworfene Probleme werden in Standard Reviewprotokoll festgehalten
- Disziplin benötigt, um Review nach entdecktem Fehler abzubrechen
→ Art der Korrektur ist nicht Teil der Review
- Reviews & Walkthroughs fördern eine Lernkultur im Betrieb

Review - Checkliste

1. Reviews im Projektplan vorsehen
 - alle Phasen mit einer Review abschliessen
 - Zeit für Vorbereitung & Sitzung planen
2. Review-Ergebnisse systematisch erfassen
 - Anzahl Fehler pro Objekt/Artefakt, aufgewendete Zeit
 - Formular verwenden, wo Beschreibung, Schweregrad & Typ der Beanstandung erfasst werden können
 - Festlegen, wer, wo und wie Metrikdaten ablegt (verantwortliche Person)
3. Den Review-Teilnehmern Rollen zuordnen
 - bspw. Autor, Moderator/Protokollführer, Leser
4. Sicherstellen, dass jeder Teilnehmer sich vorbereitet
 - bspw. Beanstandungen auf Formular vorerfasst mitbringen

9.3 Anforderungsüberprüfung in hybriden Projekten (SoDa)

Bei hybriderem Vorgehen sind nur wichtigste Anforderungen im Produktbacklog erfasst. Anforderungen werden erst im Rahmen der Sprintplanung präzisiert. Änderungen/Ergänzungen können jederzeit einfließen und Priorisierung wird laufend nachgeführt. Dazu gibt es in Scrum/SoDa drei Institutionen zur Anforderungsüberprüfung:

- **Backlog Grooming:** Priorisierung der Anforderungen, ihre Verständlichkeit und gegenseitige Abhängigkeit vom PO in regelmässiger Absprache mit Auftraggeber hinterfragen & überprüfen
- **Sprint-Abnahme:** Umsetzung der Anforderungen, laufende Änderung & Ergänzung und deren neue Abstimmung zwischen Product Owner und Auftraggeber kontinuierlich aushandeln
- **Retrospektive:** Qualität der Anforderungen hinterfragen und für künftige Sprints verbessern
→ agiles Vorgehen fördert laufenden Kontakt mit Arbeitgeber und verbessert das frühe Erkennen von Fehlern. Iterativer Ansatz reduziert Kosten von nachträglichen Korrekturen der Anforderungen.